Development of Advanced Fault Diagnosis Techniques for Complex Industrial Processes

Hongyang Yu
B.Eng (Hons), Dip.Eng(Merit)

December 2015

Submitted in fulfilment of the requirements for the Degree of
Doctor of Philosophy
National Centre for Maritime Engineering and Hydrodynamics
Australian Maritime College
University of Tasmania

Summary

Modern industrial processes are systems with a high degree of complexity. These systems comprise of a large number of components functioning in harmony to produce high quality products. In practice, the operating states of these components are monitored in real-time to determine whether there are abnormalities in process operation. There are a number of challenges associated with monitoring a large number of process components, for instance, high monitoring cost and flooding of false alarms. To address these problems, a multivariate statistical process monitoring (MSPM) framework has been developed in recent years. The MSPM performs multivariate statistical analysis on real-time process data to generate two monitoring statistics capable of identifying abnormalities in all aspects of process operations.

Many researchers have proposed a variety of techniques within the framework of MSPM. This thesis advances these developments by proposing several novel extensions in the areas of features extraction, robust online fault diagnosis and multivariate dynamic risk assessment. The first major contribution of this work is the use of Copula, a method for modelling complete dependence structure between random variables, for non-Gaussian feature extraction. The Copula method is then combined with Spearman's rank correlation coefficient for nonlinear feature extraction. Due to the use of the Spearman's correlation coefficient, the proposed technique is also robust to data contamination. Another type of technique based on Nonlinear Gaussian Belief Network is also proposed for robust feature extraction from noisy data with nonlinear variations. The second major contribution is the development of a powerful visualization tool for real-time process monitoring. This visualization tool is derived from the well-known nonlinear feature extraction algorithm, the Self-organizing Map. A direct visualization of the real-time operating state of processes is presented on a 2D map. A number of operating regions have also been identified on the 2D map, allowing for a more refined process monitoring.

The third main contribution of this thesis is the integration of the process monitoring techniques into the operational risk assessment framework. The process monitoring statistics are transformed to indicate the real-time probability of faulty conditions. In the meantime, the possible process losses due to likely fault condition are also estimated. The probability of fault and possible process losses are then combined to determine the operational risk of process. Based on the risk level, the most effective remedial measures can be easily determined.

Keywords: Multivariate statistical process monitoring, Gaussian Copula, Spearman's correlation coefficient, Kendall tau’s correlation coefficient, Self-Organizing Map, Non-linear Gaussian belief network, feature extraction, robust fault diagnosis, dynamic risk assessment.

Approvals

Doctor of Philosophy Dissertation

Approved by

Prof. Faisal I. Khan
Associate Dean (globalization)
Australian Maritime College
University of Tasmania

Vale Research Chair of Process Safety and Risk Engineering
Faculty of Engineering and Applied Science
Memorial University of Newfoundland

Signature: _____ Date: 15/04/2016

Dr. Vikram Garaniya
Course Coor**d**inator
National Centre for Maritime Engineering and Hydrodynamics
Australian Maritime College
University of Tasmania

Signature: _____ Date: 15/04/2016

Declaration and Statements

Declaration of Originality

I declare that this is my own work and has not been submitted in any form for another degree or diploma at any university or other institution of tertiary education. Information derived from the published or unpublished work of others has been duly acknowledged in the text and a list of references if given.

Authority of Access

This thesis may be made available for loan and limited copying in accordance with the Copyright Act 1968.

Statement Regarding Published Work Contained in the Thesis

The publishers of the papers comprising Chapters 2 to 8, inclusive, hold the copyright for that content, and access to the material should be sought from the respective journals. The remaining non published content of the thesis may be made available for loan and limited copying and communication in accordance with the above Statement of Access and the Copyright Act 1968.

Signature: _____ Date: 05/01/2016

Thesis by Journal Articles

The following six published journal articles constitute the content of this thesis.

- Chapter 2: Yu, H., Khan, F., & Garaniya, V. (2015). A probabilistic multivariate method for fault diagnosis of industrial processes. *Chemical Engineering Research and Design, 104*, 306-318. doi:10.1016/j.cherd.2015.08.026.

- Chapter 3: Yu, H., Khan, F., Garaniya, V. (2015). A sparse PCA for nonlinear fault diagnosis and robust feature discovery of industrial processes, accepted in *AICHE Journal*. doi: 10.1002/aic.15136.

- Chapter 4: Yu, H., Khan, F., & Garaniya, V. (2015). Nonlinear Gaussian Belief Network based fault diagnosis for industrial processes. *Journal of Process Control, 35*, 178-200. doi:10.1016/j.jprocont.2015.09.004.

- Chapter 5: Yu, H., Khan, F., & Garaniya, V. (2015). Modified Independent Component Analysis and Bayesian Network-Based Two-Stage Fault Diagnosis of Process Operations. *Industrial & Engineering Chemistry Research*, *54*(10), 2724-2742. doi:10.1021/ie503530v.

- Chapter 6: Yu, H., Khan, F., & Garaniya, V. (2015). Risk-based fault detection using Self-Organizing Map. *Reliability Engineering & System Safety*, *139*, 82-96. doi:10.1016/j.ress.2015.02.011.

- Chapter 7: Yu, H., Khan, F., & Garaniya, V. (2015). Risk-based Process System Monitoring using Self-organizing Map integrated with Loss Functions, accepted in *The Canadian Journal of Chemical Engineering*.

Journal articles not included in this thesis, but can be accessed as supplementary material.

- Yu, H., Khan, F., & Garaniya, V. (2014). Self-Organizing map based fault diagnosis technique for non-Gaussian processes, *Industrial & Engineering Chemistry Research*, 53 (21) pp. 8831-8843. doi:10.1021/ie500815a.

- Yu, H., Khan, F., & Garaniya, V. (2015). An alternative formulation of PCA for process monitoring using distance correlation, accepted in *Industrial & Engineering Chemistry Research*. doi: 10.1021/acs.iecr.5b03397.

Co-Authorship for all Journal Articles

Conceived idea and designed the Case Studies: Khan, Yu and Garaniya
Performed the Case Studies: Yu
Analysed the data: Yu
Wrote the manuscript: Yu
Manuscript evaluation and submission: Khan and Garaniya

Prof. Faisal I. Khan
Associate Dean (globalization)
Australian Maritime College
University of Tasmania

Vale Research Chair of Process Safety and Risk Engineering
Faculty of Engineering and Applied Science
Memorial University of Newfoundland

Signature: _____ Date: 05/01/2016

Dr. Vikram Garaniya
Course Coordinator
National Centre for Maritime Engineering and Hydrodynamics
Australian Maritime College
University of Tasmania

Signature: _____ Date: 05/01/2016

Mr. Hongyang Yu
National Centre for Maritime Engineering and Hydrodynamics
Australian Maritime College
University of Tasmania

Signature: _____ Date: 05/01/2016

Acknowledgement

I would like to express my deepest gratitude to the following people who helped develop this thesis:

- Prof. Khan and Dr. Garaniya for your guidance, encouragement and support throughout my PhD study;
- Ping Yu and Yuanchong Hong for being the most understanding parents;
- Joycelyn Woo for standing by my side through many difficult periods;
- And finally, all my friends for their companionship.

To Ping Yu, Yuanchun Hong and Joycelyn Woo.

List of Symbols

$\mathbf{I}_n$	Identify matrix of dimension $n \times n$
$\mathbb{R}^n$	Euclidean *n*-dimensional space
$\mathbb{Z}^+$	The set of natural numbers (positive integers)
$\mathbf{X}_j^i$	Entry of the matrix **X** in the i^{th} row and j^{th} column
$p(\mathbf{x})$	Probability density function of **x**
$P(\mathbf{x})$	Probability distribution function of **x**
$P(\mathbf{x}\|\mathbf{y})$	Conditional probability distribution of **x** given **y**
$P(\mathbf{x}, \mathbf{y})$	Joint probability distribution of **x** given **y**
$\mathbf{x} \sim p(\mathbf{x})$	x is sampled from $p(\mathbf{x})$
$\mathcal{O}(N)$	The computational complexity is order *N* operation

List of Mathematical Operators

$\mathbf{X}^T$	Transpose of matrix **X**
$\mathbf{X}^{-1}$	Inverse of matrix **X**
$\mathrm{tr}(\mathbf{X})$	Trace of **X**
$\|\|\mathbf{X}\|\|$	Matrix gain or matrix norm
$\mathrm{triu}(\mathbf{x})$	Converting a row vector **x** into an upper triangular matrix with all diagonal elements equal to zero
$\|\|\mathbf{X}\|\|_0$	l_0 norm of **x**
$\|\|\mathbf{X}\|\|_1$	l_1 norm of **x**
$\|\|\mathbf{X}\|\|_2$	l_2 norm of **x**
$\mathrm{card}(\mathbf{x})$	Cardinality of **x**
$\mathrm{supp}(\mathbf{x})$	Support of **x**
$\mathbb{E}(\mathbf{x})$	Expectation of random variable **x**
$\min_{\mathbf{x}} f(\mathbf{x})$	Minima with respect to **x**
$\max_{\mathbf{x}} f(\mathbf{x})$	Maxima with respect to **x**
$\mathrm{argmin}_{\mathbf{x}} f(\mathbf{x})$	The argument **x** that minimizes the operand
$\mathrm{argmax}_{\mathbf{x}} f(\mathbf{x})$	The argument **x** that maximizes the operand

List of Abbreviations

BN	Bayesian Network
CPDF	Conditional probability density functions
CPT	Conditional probability table
EM	Expectation-Maximization
FAR	False alarm rate
FDR	Fault detection rate
MSPM	Multivariate statistical process monitoring
NLGBN	Non-linear Gaussian Belief Network
i.i.d	Independent Identically Distributed
PCA	Principal Component Analysis
ICA	Independent Component Analysis
MICA	Modified Independent Component Analysis
MLE	Maximum Likelihood Estimation
KPCA	Kernel PCA
KICA	Kernel ICA
T^2	Hotelling's statistic
SPE	Squared prediction error
s.t.	Subject to

Table of Contents

1 Introduction

1.1 Aim of the Research

Determining the real-time states of process operation through directly monitoring a large number of critical process components can be difficult. In many cases, a fault condition only causes subtle disturbances of the process operation which are difficult to be identified by simply looking at the behaviour of individual components. These subtle disturbances, if not corrected promptly, can result in complete failure of the system. In addition, the process components are highly integrated; the malfunction of a small component can ripple throughout the entire system leading to multiple upsets and flooding of alarms. In fact, the flooding of alarms can easily conceal the true root-cause of the malfunction. In consequence, process plants might need to be shut down for a long period of time for overhaul, incurring significant capital losses. The multivariate statistical process monitoring (MSPM) effectively addresses these problems by extracting the latent features of the process operation. In MSPM, each process component is considered as a random variable. The monitored data associated with all the process variables are projected into a subspace with lower dimensionality in which the latent features of the process operation are preserved. The process variation is then quantified and monitored within this subspace. A large increase in magnitude of process variation indicates abnormal process operation.

The Principal Component Analysis (PCA) is arguably the most widely applied statistical feature extraction technique for process monitoring. The subspace of PCA is spanned by the unit vectors representing directions of maximum variance in the high dimensional process data. These unit vectors are also known as the Principal Components (or Eigenvectors) and are orthogonal to each other. A number of techniques (and algorithms) have been proposed to extract the principal components from the process data. The Singular Value Decomposition (SVD) is by far the popular eigenvector extraction technique. In the standard SVD procedure, the correlation matrix of the process data, based on the Pearson's correlation coefficient, is first determined. The principal components are then extracted as the eigenvectors of the correlation matrix. The total number of eigenvectors is the same as the number of process variables. The amount of variance each eigenvector captures is in fact equal to its corresponding eigenvalue. For process monitoring, a smaller number of eigenvectors that capture more than 70% of data variance are retained to construct the subspace. The correlation structures of the process variables constituting the latent features of the process operation are preserved in this subspace. In this respect, this subspace is also referred as the latent space. During real-time monitoring, the real-time process data samples are projected into the latent space. The process real-time variation within the latent space is the systematic variation of the process operation and is measured by a statistic called the Hotelling's T^2 statistic. On the other hand, the non-systematic variation or the residual variation is measured by the SPE (Squared Prediction Error) statistic. The detailed procedure of generating these two

statistics and their process monitoring mechanisms are explained in details in the later section of this Thesis.

There are three major issues associated with PCA-based process monitoring. First, the Pearson's correlation coefficient is only able to model linear relationship between process variables. For non-linear relationships, such as $y = x^2$, the Pearson's correlation coefficient fails completely in modelling. However, non-linear relationships between process variables are prevalent in modern industrial processes. The second major issue is also closely related to the linearity assumption of the Pearson's correlation coefficient. In order for the principal component to represent directions of maximum variance, it has to be one of the major axes of a perfect eclipse encompassing the process data. This necessitates the process data follow perfect Gaussian distribution. The operation of modern industrial processes is constantly subjected to systematic or external disturbances. Many of the process variables are also closed-loop controlled with PID controllers. Therefore, the process data collected under these unstable conditions rarely follow Gaussian distribution. The third major issue comes from the fact that the Pearson's correlation coefficient is the second moment of process data, which is not robust to data contamination at all. A small number of data outliers can lead to extraction of eigenvector that do not represent directions of maximum variance. These three limitations can seriously affect the process monitoring performance of the standard PCA-based techniques. The main focus of this research work is to improve the efficacy of the PCA-based process monitoring technique by addressing its three major limitations.

1.2 Research Milestones and Research Questions

The development of an effective and robust process monitoring technique involves accomplishing the following milestones.

- Development of a latent space model capable of retaining non-Gaussian features in the latent space for process monitoring;
- Incorporation of non-linear correlation measures into the non-Gaussian latent space model to capture nonlinear relationships between process variables;
- Modify the developed non-linear and non-Gaussian latent space model to enable robust feature extraction by adopting robust statistics;
- Integration of the developed process monitoring model with multivariate loss modelling techniques to build a statistics-based dynamic risk assessment framework for complex industrial process systems.

In addition, each milestone is achieved by answering a relevant research question. These questions are listed below.

- Is it feasible to develop computationally tractable methods for feature extraction when process data are modelled using non-Gaussian distributions?
- Which is the most suitable non-linear correlation measure for modelling nonlinear relationships among process variables of complex industrial processes?
- How robust is this correlation measure against data contamination, if it is computed using robust statistics?
- Is it possible to compute the probability of failure based on the monitoring statistics of the developed latent space model? What is the best to integrate probability of failure with real-time loss to determine the dynamic risk of process operation?

1.4 Scope and Contributions

A great deal of effort has been devoted to address the three major issues of PCA-based process monitoring in this Thesis. To deal with the first and the third major issues, three different types of techniques have been proposed to enable non-linear and robust feature extraction for process monitoring. The Pearson's correlation matrix is replaced with the Spearman's correlation matrix which is able to model non-linear monotonic relationships and is robust to data contamination. The eigenvectors extracted from the Spearman's correlation matrix captures more information regarding the systematic variation of the process operation, thereby significantly improving the process monitoring performance. In addition, another class of non-linear feature extraction technique based on the Non-linear Gaussian Belief Network (NLGBN) is also proposed for process monitoring. As compared to the linear subspace projection of PCA, NLGBN introduces an additional layer of non-linear activation functions between the process variable and the subspace to model non-linear correlation structure. Likewise, this allows more information to be retained in the subspace to produce better process monitoring result. Finally, a Self-Organizing Map (SOM) based process monitoring technique is developed to achieve a more intuitive and refined process monitoring. The process operation is represented as a dynamic trajectory on a 2D map which provides a direct visualization of the dynamic behaviour of process under normal and fault conditions. The 2D map is also classified into different regions corresponding to different degrees of fault progression for more efficient determination of the most appropriate remedial measures.

The Pearson's correlation coefficient is also known as the Pearson's product-moment coefficient which can be considered as a second-order statistic capturing only pairwise correlation structures in the process data. Gaussian distribution happens to be perfect distribution to explain data variance where only pairwise correlation structure exists. To explain the non-Gaussian variance, a high-order statistic which captures the complete dependence structure between process variables is necessary. The Copula is one of the best candidates for modelling complete dependence structure. A Copula consists of two major elements: the univariate marginal distributions of the process variables and a copula density function. The copula function is a ratio between the (1) joint probability density function and (2) the joint probability density function under complete independence. In this respect, the complete dependence structure of the process variables can be captured in copula density function as opposed to only second-order dependence structure captured in PCA. This implies that Copula is able to model non-Gaussian probability distribution. For process monitoring, a Copula distribution (rather than a Gaussian distribution) is fitted to explain the variance of normal process operation. An upper probabilistic control limit is defined over the normal operation region enabling more accurate abnormality detection.

In addition to addressing the major issues of standard PCA-based process monitoring, this thesis proposes novel techniques to integrate MSPM into the framework of operational risk assessment. To obtain the process operational risk, the real-time probability of fault condition and the instant process losses are to be quantified. The real-time probability of fault is estimated using two SOM-based approaches. The first

approach defines an upper control limit for the dynamic trajectory as three standard deviations from the centre of normal operation. The probability of fault can be easily calculated, using standard statistical method, as how likely the dynamic trajectory will exceed the upper control limit. The second approach estimates the probability of fault using a non-parametric density estimation method specifically developed for SOM. Upon obtaining the probability of fault, the instant process losses are modelled using a series of loss functions. Finally, the real-time risk of operation is computed as the product between the probability of fault and the instant process losses. As compared to the conventional dynamic risk assessment, the proposed risk estimation method is multivariate and considers the non-linear relationships between process variables. Therefore, it provides a more realistic risk assessment.

The implementation of the developed techniques could offer major benefits to operation of complex industrial process systems. Real-time operating states of the process system can be determined in real-time with minimum delay and high accuracy. This piece of information can be used by Engineers or operators to determine the best operational setups in a much efficient manner to maximize production efficiency. Meanwhile, anomalies in operation can be detected and diagnosed timely to allow prompt implementation of appropriate remedial measures. As a result of these improvements, the overall reliability of process operation is increased and the likelihood of catastrophic failures with substantial losses is minimized.

1.6 Thesis Organization

A summary of the thesis outline is provided in below section. These chapters are, to a large extent, self-contained and can be read independently.

Chapter 2: A Probabilistic Multivariate Method for Fault Diagnosis of Industrial Processes

This chapter introduces the use of Gaussian Copula for modelling complete variable dependence structure and non-Gaussian process variation. A probabilistic process monitoring scheme including on-line fault detection and diagnosis is developed.

Chapter 3: A sparse PCA for nonlinear fault diagnosis and robust feature discovery of industrial processes

Chapter 3 explores the robust and non-linear relationships modelling features of Spearman's rank and Kendall tau's correlation measures for process monitoring. The process data is transformed through a semi-parametric Gaussian transformation to be Gaussian distrusted, whereby standard PCA feature extraction can be effectively applied. The eigenvectors are extracted from the robust correlation matrix to retain non-linear process variation in the subspace. The T^2 and SPE statistics for process monitoring are computed the same way as conventional PCA.

Chapter 4: Nonlinear Gaussian Belief Network Based Fault Diagnosis for Industrial Processes

Chapter 4 proposes an online fault diagnosis technique based on Non-linear Gaussian Belief Network. A three-layer NLGBN is constructed to extract non-linear features from noisy process data. The parameters of NLGBN are trained using a variational Expectation-Maximization algorithm. A single statistic is generated at the top layer of the NLGBN for process monitoring.

Chapter 5: Self-Organizing map based fault diagnosis technique for non-Gaussian processes

Chapter 5 develops a powerful visualization tool for process monitoring based on the Self-Organizing Map, a neural network type non-linear feature extraction tool. The dynamical behaviour of the process under normal or faulty operations is presented as a dynamic trajectory on a 2D map. The deviation of the trajectory from the centre of normal operation is monitored for online abnormality detection.

Chapter 6: Modified Independent Component Analysis and Bayesian Network based Two-stage Fault Diagnosis of Process Operations

Chapter 6 deals with a specific issue of data unavailability for multivariate statistical process monitoring. The on-line fault detection and diagnosis is broken into two stages. The first stage uses the standard MSPM for process monitoring with available data. The second stage brings the information from the first monitoring stage into Bayesian Network for further inference. The Bayesian network then locates the true root-cause of fault from the process variable without measurement data.

Chapter 7 & 8: Process operational risk assessment

Both chapters 7 and 8 focus on integrating the MSPM into the dynamic risk assessment framework. The SOM is used for fault detection and quantification of probability of fault. The instant process losses are estimated using a series of loss functions. The operational risk is calculated as the product between the probability of fault and instant process losses.

Chapter 9: Conclusions

This final chapter summarizes the major finds of this thesis and point out several new directions for future research.

Appendices

The appendices contain derivations, proofs of theorems and other ancillary information.

2 A Probabilistic Multivariate Method for Fault Diagnosis of Industrial Processes

Abstract

A probabilistic multivariate fault diagnosis technique is proposed for industrial processes. The joint probability density function containing essential features of normal operation is constructed considering dependency among the process variables. The dependence structures are modelled using Gaussian copula. The Gaussian copula uses rank correlation coefficients to capture the nonlinear relationships between process variables. For real-time monitoring, the probability of each online data samples is computed under the joint probability density function. Those samples having probabilities violating a predetermined control limit are classified to be faulty. For fault diagnosis, the reference dependence structures of the process variables are first determined from normal process data. These reference structures are then compared with those obtained from the faulty data samples. This assists in identifying the root-cause variable(s). The proposed technique is tested on two case studies: a nonlinear numerical example and an industrial case. The performance of the proposed technique is observed to be superior to the conventional statistical methods, such as PCA and MICA.

Keywords: Fault detection, Fault diagnosis, nonlinear processes, Multivariate Gaussian Copula.

2.2 Introduction

Modern industrial processes are designed to handle a number of simultaneous tasks to achieve maximum production efficiency. The interactions among these components are dynamic, often subjected to disturbances and a high degree of nonlinearity. To determine the real-time operating states of a process, a set of process variables associated with the crucial components are monitored online. Online process data from the monitored process variables are analysed using advanced multivariate statistical techniques for early fault detection and deducing of the root-cause(s). These multivariate statistical techniques determine a small number of latent variables which capture the correlation structure of the original process variables.[1-4] Typically, any process data sample that compromises the integrity of the correlation structure is flagged as a faulty sample and the process variable(s) contributing the most to this is identified as the root-cause(s).[5,6]

The most commonly applied multivariate statistical techniques include the Principle Component Analysis (PCA) and Partial Least Square (PLS).[7-10] These two methods rely heavily on three simplifications of the latent features of the industrial processes: (1) the process variables are assumed to have Equal and infinitesimal variance; (2) each latent variable approximately follows zero mean Gaussian distribution; (3) each latent variable is a weighted linear combination of all the process variables.[11] These three simplifications establish the necessary conditions for performing Singular Value Decomposing (SVD) on the process data to determine the optimal orthogonal projection weight vectors (loading vectors). However, the operation of the modern processes is constantly subjected to external and systematic disturbances, which lead to generation process data containing highly nonlinear and non-Gaussian features. PCA/PLS might not be able to extract enough latent features to conduct robust fault detection and diagnosis. In addition, another major weakness of these two methods lies in fact that the control limits encompassing the normal region of process operation are also derived from Gaussian distribution.[12,13] This could lead to serious misclassification issue if the faulty process data samples are not linearly separable from the normal process data samples in the low dimensional feature space.[14]

To capture the non-Gaussian latent features of the process operation, the Independent Component Analysis (ICA) based feature extraction technique is proposed.[1,15] The latent variables of the ICA can retain the non-Gaussian latent features of the process operation. This is achieved by determining a set of non-orthogonal projection weight vectors that maximize the negative entropy or Kurtosis following the independent component direction of each latent variable.[16,17] The ICA addresses one of the weaknesses of the PCA/PLS based methods; the process variables are still assumed to be noise free and are linearly related to the latent variables. The Kernel extension of the ICA, known as KICA, is developed in modelling the nonlinear relationships between the process variables and the non-Gaussian latent variables.[18] The KICA introduces an extra step of data pre-processing. The original process data is first mapped from the relatively low dimensional measurement space to a theoretically infinite dimensional space (when the radial basis kernel is used). According to Vapnik-Chervonenkis theory, process data samples that cannot be linearly classified often become linearly separable in a higher dimensional

space.[19] By applying ICA to the transformed process data in a high dimensional space, the faulty process data samples used to be linearly inseparable in the original low dimensional measurement space can be easily classified, leading to high fault detection rate. However, the high-dimensional mapping of the process data is one way which implies that it is impossible to compute the contribution of each process variable through reverse projection.[1,20] Due to this reason, fault diagnosis technique based on KICA is yet to be developed. Additionally, the high-dimensional mapping also significantly increases the computational time.

Multivariate method using Copula could serve as an efficient alternative to traditional statistical methods in process monitoring. Copula is a powerful method to model joint probability distribution of multiple random variables. The joint probability distribution is expressed in terms of the univariate marginal distributions of the random variables and a copula function.[21-23] The derivative of the copula function (copula density function) defines the strength of dependence among the random variables. In fact, it is a ratio between the (1) joint probability density function and (2) the joint probability density function under complete independence. In this respect, the complete dependence structure of the process variables can be captured in copula density function as opposed to only second-order dependence structure captured in PCA. This implies that Copula is able to model non-Gaussian probability distribution. Additionally, in the copula density function, the correlation between each pair of process variable is described using a nonparametric statistic, known as the Spearman's (rank) correlation coefficient.[24] The Rank coefficient is a statistical measure of the strength of the monotonic relationship between any two process variables.[25] As compared to Pearson's coefficient used in PCA and ICA[*] which only models linear correlation between process variables, Rank correlation is more flexible in handling extremely nonlinear and monotonic relationships. Furthermore, Copula is able to define a semi-parametric (parametric copula function and nonparametric Rank correlation) joint probability density function over the process data samples. The noise in the process data can be easily accommodated.

In this work, a specific type of copula, known as the Gaussian Copula, is used to model the probability density function of the normal process data. The parametric Gaussian copula is chosen as it has less model parameters as compared to other parametric copulas. This reduces the number of free parameters for optimization. Since Copula is only able to model nonlinear monotonic relationships, the process variables are first monotonized using Rolling Pin method.[26] During offline training, the copula model is adapted to the monotonized normal process data using Maximum Likelihood Estimation. The optimal copula model is adopted to form the joint probability density function over the monotonized normal process data. A probabilistic control limit for normal operation is also defined over the same region. During online monitoring, online process data samples are first monotonized and their probabilities under the joint probability density function are computed. The online samples with a probability smaller than the control limit are flagged as faulty samples. For fault diagnosis, the dependence

[*] ICA also uses Pearson's correlation in the data whitening step

structure of the online data samples is compared with that of the normal. The variables that contribute the most to the breakdown of the structure are identified as root-causes.

The remainder sections of the paper are organized as follows. The fundamental concept of the Copula function is briefly introduced in Section 2.3. The complete fault detection and diagnosis methodology using copula method is explained in Section 2.4. In Section 2.7, the proposed method is then tested on two case studies including a motivational case study and an industrial case study. In addition, the performance of the proposed method is also compared with the performances of PCA and ICA. Finally, the conclusions and potential area for future development of this work are presented in Section 2.8.

2.3 Preliminaries

In order to establish a consistent style of presentation, the following notations are first introduced. In this article, a set of d continuous process variables are arranged in a row vector $\mathbf{X} = [X_1, X_2, \dots, X_d], \mathbf{X} \in \mathbb{R}^d$. The numerical value of each process variable is represented using small letter notation $x_i \in \mathbb{R}, i \in \{1, \dots, d\}$. A single process data sample is a row vector containing the instant numerical values of each process variable $\mathbf{x}^j = \left[x_1^j, x_2^j, \dots, x_d^j\right], \mathbf{x}^j \in \mathbb{R}^d, j \in \{1, \dots, n\}$, where n is the total number of samples. The probability density function (PDF) of a process variable X_i is defined as $f(X_i)$. similarly, the cumulative distribution function (CDF) of X_i is defined as $F(X_i)$. In fact, $F(X_i)$ is a probability integral transformation of X_i, which has a uniform distribution in the range [0,1]. It is noted that when a numerical value x_i is given, there exists $f(x_i) = p(X_i = x_i)$ and $F(x_i) = P(X_i \leq x_i)$. Let $U_i = F(X_i)$ and $u_i = F(x_i)$, the following condition holds as U_i is uniformly distributed in [0,1].

$$P(U_i \leq u_i) = u_i \tag{2.1}$$

Since $u_i = F(x_i)$, $x_i = F^{-1}(u_i)$. It is evident that following Equality is true.

$$P(U_i \leq u_i) = P\left[X_i \leq F^{-1}(u_i)\right] = u_i \tag{2.2}$$

The joint cumulative probability distribution of $\mathbf{X}$ can be expressed in terms of Eq. (2.1) and Eq. (2.2) as following.

$$F(x_1, x_2, ..., x_d) = P\left[U_1 \leq F(x_1), U_2 \leq F(x_2), ..., U_d \leq F(x_d)\right] \tag{2.3}$$

According to Sklar's theorem[27], for a multivariate joint probability distribution such as Eq.(2.3) with uniformly distributed marginal CDFs, there exists a copula function $C: [0,1]^d \rightarrow [0,1]$ such that.

$$\begin{gathered} C\left[F(x_1), F(x_2), ..., F(x_d)\right] = P\left[U_1 \leq F(x_1), U_2 \leq F(x_2), ..., U_d \leq F(x_d)\right] \\ F(x_1, x_2, ..., x_d) = C\left[F(x_1), F(x_2), ..., F(x_d)\right] \end{gathered} \tag{2.4}$$

In this regard, the joint probability distribution is decomposed into its univariate marginal CDFs and a copula function. The copula function contains the information regarding the complete dependence structure of **X**. This is easily shown by taking derivative on both sides of Eq. (2.4).

$$f(x_1, x_2, ..., x_d) = c\left[F(x_1), F(x_2), ..., F(x_d)\right]\prod_{i=1}^{d} f(x_i) \tag{2.5}$$

where c is the copula density function $c: [0,1]^d \rightarrow \mathbb{R}^+$. Eq. (2.5) can be rewritten as;

$$\frac{f(x_1, x_2, ..., x_d)}{\prod_{i=1}^{d} f(x_i)} = c\left[F(x_1), F(x_2), ..., F(x_d)\right] \tag{2.6}$$

The copula density is a ratio between the (1) joint probability density function and (2) the joint probability density function under complete independence. The complete dependence structure of the process variables can be captured in copula density function.

2.4 Methodology

2.4.1 Monotonization of process variables

Copula is only able to capture nonlinear monotonic relationships between process variables. This requires that the relationship within any pair of process variables to be strictly monotonic. However, this is often not achievable for many process operations, in particular for those subjected to high-order variations; one process variable could have a quadratic relationship with another rather than a monotonic relationship. In order to address this issue, the Rolling Pin method proposed by Mohseni Ahooyi, et al.[26] is used to monotonize the relationships in every pair of process variables. A strictly-increasing monotonic relationship between two process variables $(X_i, X_k), i \neq k, i, k \in \{1, \dots, n\}$ should satisfy.

$$\frac{\partial X_k}{\partial X_i} > 0 \tag{2.7}$$

If this condition is violated, the following transformation can be used to monotonize the process variables.

$$Y_i = (1 - a_i)X_i + a_i X_r \tag{2.8}$$

where a_i is the transformation parameter, $a_i \in [0,1]$, and X_r is the reference variable for transformation. The reference variable is carefully selected such that the transformed process variable Y_i has a strictly-increasing relationship with respect to X_r. In the work of Mohseni Ahooyi, et al.[26], three selection methods of the reference variable are discussed. In this study, the witness approach is adopted and the reference variable has a Gaussian

distribution with zero mean and unit variance. It can be shown that, after transformation, the relationship within each pair of process variables is strict-monotonically increasing.

$$\frac{\partial Y_i}{\partial X_r} > 0 \Leftrightarrow \frac{\partial X_i}{\partial X_r} + \frac{a_i}{1-a_i} > 0 \tag{2.9}$$

$$\begin{aligned} \frac{\partial Y_k}{\partial Y_i} &= (1-a_i)\frac{\partial X_i}{\partial Y_i} + a_i \frac{\partial X_r}{\partial Y_i} \\ &= (1-a_i)\frac{\partial X_i}{\partial X_r}\frac{\partial X_r}{\partial Y_i} + a_i \frac{\partial X_r}{\partial Y_i} \\ &= (1-a_i)\left(\frac{\partial X_i}{\partial X_r} + \frac{a_i}{1-a_i}\right)\frac{\partial X_r}{\partial Y_i} > 0 \end{aligned} \tag{2.10}$$

It is noticed that the transformation from X_i to Y_i is injective. Suppose $y_i^a = y_i^b$ for some arbitrary values of the i^{th} process variable $x_i^a, x_i^b \in \mathbf{x}$ and x_r is the same for both x_i^a and x_i^b . Then.

$$\begin{aligned} (1-a_i)x_i^a + a_i x_r &= (1-a_i)x_i^b + a_i x_r \\ \Rightarrow x_i^a &= x_i^b \end{aligned} \tag{2.11}$$

In addition, for a set of continuous process variables $\mathbf{X} = [X_1, X_2, \dots, X_d], \mathbf{X} \in \mathbb{R}^d$ and the transformed process variables $\mathbf{Y} = [Y_1, Y_2, \dots, Y_d], \mathbf{Y} \in \mathbb{R}^d$, the determinant of the Jacobian matrix $\mathbf{J} = \frac{\partial \mathbf{Y}}{\partial \mathbf{X}}$ is given as.

$$\det(\mathbf{J}) = \prod_{i=1}^{d-1}(1-a_i) > 0 \tag{2.12}$$

Based on the property of the injective transformation and a non-zero determinant, the following Equality condition is true.[28]

$$f(\mathbf{X}) = f(\mathbf{Y})\,|\det(\mathbf{J})| \tag{2.13}$$

Taking integration on both sides of Eq. (2.13), the joint cumulative probability distribution of the transformed variables is obtained as

$$F(\mathbf{Y}) = \int_{\mathbf{Y}} f(\mathbf{Y}) d\mathbf{Y} \tag{2.14}$$

Substituting Eq.(2.13) into (2.14) and also replacing $d\mathbf{Y}$ with $|\det(\mathbf{J})|d\mathbf{X}$ yields.

$$\begin{aligned} F(\mathbf{Y}) &= \int_{\mathbf{Y}} \frac{1}{|\det(\mathbf{J})|} f(\mathbf{X}) |\det(\mathbf{J})| \, \mathrm{d}\mathbf{X} \\ &= \int_{\mathbf{X}} f(\mathbf{X}) \, \mathrm{d}\mathbf{X} \\ &= F(\mathbf{X}) \end{aligned} \tag{2.15}$$

This implies that the joint probability distribution function of the transformed process variables is Equivalent to that of the original process variables.

$$F(x_1, x_2, ..., x_d) = F(y_1, y_2, ..., y_d) \tag{2.16}$$

Therefore, the copula function in Eq.(2.4) can be reformatted as.

$$F(x_1, x_2, ..., x_d) = C\left[F(y_1), F(y_2), ..., F(y_d)\right] \tag{2.17}$$

Since the relationship within each pair of the transformed process variables is strict-monotonically increasing, the copula function in Eq. (2.17) captures the complete dependence structure (monotonic and non-monotonic) of the process variables.

2.5 Offline Training

The objective of the off training step is to obtain a joint probability density function over a predefined number of normal process data samples. This joint probability density function contains essential features of the normal process operational variations. A robust probabilistic control limit can then be defined for the probability density function to discern the faulty process data samples during online monitoring. The parameters estimated during the training process are the transformation parameters a_i. Typically, any parametric copula with symmetric correlation matrix can be used for the transformed process variables.[26] In this case, Gaussian copula is chosen because it has less number of model parameters as compared to other parametric copulas. This reduces the computational time for estimation.

In Eq. (2.8) the transformation parameter a_i provides a trade-off between how much information is inherited from X_i and the strength of monotonic relationship concerning Y_i and X_r. Specifically, if a_i has a value extremely close to 1, the transformed process variables have strong monotonicity but contain little information regarding the process variation. Conversely, if a_i is too small, the strict- monotonically increasing relationship within each pair of the transformed process variables cannot be ensured, leading to inaccurate modelling of the joint probability density function. Maximum Likelihood estimation is an ideal technique to estimate the appropriate values of a_i. The estimated a_i constructs a probability density function that best explains the variation of the normal process data. Naturally, the information loss due to the transformation in the modelling process is minimized by incrementally increasing the likelihood of generating the same training data. The joint probability density function of the transformed process variables is obtained by taking the derivative on both sides of Eq.(2.17).

$$f(x_1, x_2, ..., x_d) = c[F(y_1), F(y_2), ..., F(y_d)] \prod_{i=1}^{d} f(y_i)(1-a_i) \tag{2.18}$$

Given *n* samples of training data, the likelihood function is derived from Eq. (2.18) as.

$$\begin{aligned} L(\mathbf{a} \mid \mathbf{D}) &= f(\mathbf{D} \mid \mathbf{a}) = \prod_{j=1}^{n} c[F(y_1{}^j), F(y_2{}^j), ..., F(y_d{}^j)] \prod_{i=1}^{d} f(y_i{}^j)(1-a_i) \\ y_i{}^j &= (1-a_i)x_i^j + a_i x_r^j \end{aligned} \tag{2.19}$$

where $\mathbf{a} = \{a_1, \ldots, a_d\}$, $\mathbf{D}$ is the training data matrix, $\mathbf{D} \in \mathbb{R}^{n \times d}$. In general, the log-likelihood function is used for Maximum Likelihood Estimation. The log-likelihood function is expressed as.

$$\begin{aligned} \log\left[L(\mathbf{a} \mid \mathbf{D})\right] &= \sum_{j=1}^{n} c[F(y_1{}^j), F(y_2{}^j), ..., F(y_d{}^j)] \\ &+ \sum_{j=1}^{n} \sum_{i=1}^{d} \log\left[f(y_i^j) \right] + n \sum_{j=1}^{d} \log(1-a_i) \end{aligned} \tag{2.20}$$

The univariate CDFs, $F(y_i^j)$, and the univariate PDFs, $f(y_i^j)$, of the transformed process variables can be estimated using nonparametric Kernel density estimator.

$$F(y_i^j) = \frac{1}{nh} \int_{-\infty}^{y_i^j} \sum_{j=1}^{n} K\left(\frac{t - y_i^j}{h} \right) dy_i \tag{2.21}$$

$$f(y_i^j) = \frac{\partial F(y_i^j)}{\partial y_i} \tag{2.22}$$

Additionally, the correlation matrix for the Gaussian copula can also be directly computed from training data.

$$\rho\left[F(Y_i), F(Y_k)\right] = \frac{\mathrm{Cov}\left[F(Y_i), F(Y_k)\right]}{\mathrm{var}\left[F(Y_i)\right] \mathrm{var}\left[F(Y_k)\right]}, \forall i, k \in \{1, ..., d\} \tag{2.23}$$

It should be noted, in Eq.(2.23), the Pearson's correlation between the probability integral transformations ($F(Y_i)$ and $F(Y_k)$) is in fact the Spearman's rank correlation of the transformed process variables.[29] This enables copula to model extremely nonlinear relationships in complex industrial processes.

$$\rho_{rank}(Y_i, Y_k) = \rho\left[F(Y_i), F(Y_k)\right] \tag{2.24}$$

The rank correlation matrix is computed numerically from data. It might be ill-conditioned and non-positive semi-definite, i.e. its minimum eigenvalue equals to zero.

The following numerical optimization method is proposed to determine the minimal perturbations that make the correlation matrix positive semi-definite. Suppose an ill-conditioned correlative matrix $\boldsymbol{\rho} \in \mathbb{R}^{d\times d}$ whose minimum eigenvalue is zero. Let $\mathbf{z} = [z_1, z_2, \dots, z_m], m = \sum_{l=1}^{d-1} l$ be the perturbations introduced to both the upper triangular and lower triangular elements (excluding the diagonal elements). The perturbations have to be symmetric as the correlation matrix is also symmetric. The perturbations are not introduced to the diagonal elements as they represent self-correlations which have to be unity. After the perturbations are introduced, the new correlation matrix is updated as.

$$\hat{\boldsymbol{\rho}} = \boldsymbol{\rho} + \mathrm{triu}(\mathbf{z}) + \mathrm{triu}(\mathbf{z})^{\mathrm{T}} \tag{2.25}$$

where $\mathrm{triu}(\cdot)$ converts a row vector into an upper triangular matrix with all diagonal elements Equal to zero. Then, the objective function for optimization is the maximum element-wise percentage error between the updated correlation matrix and the original matrix.

$$obj = \max\left\{ \mathrm{abs}\left[\frac{\left(\hat{\rho}_{i,k} - \rho_{i,k}\right)}{\rho_{i,k}} \right] \times 100\% \right\}, \forall i,k \in \{1,\dots,d\} \tag{2.26}$$

where $\hat{\rho}_{i,k}$ and $\rho_{i,k}$ is the i,k^{th} element of $\hat{\boldsymbol{\rho}}$ and $\boldsymbol{\rho}$, respectively. After update, each element of the updated matrix has to be in the range [-1,1]. The above problem can then be formatted into a constrained optimization problem with a nonlinear inequality constraint and a box constraint.

$$\begin{aligned} &\min_{\mathbf{z}} obj \\ &s.t. \quad -1 < \hat{\rho}_{i,k} < 1 \\ &\quad \min\left[\mathrm{eig}(\hat{\boldsymbol{\rho}})\right] > 0 \end{aligned} \tag{2.27}$$

After the well-conditioned rank correlation matrix is obtained, the active set method [30,31] is adopted to maximize Eq. (2.20) under the box constraint $0 < a_i < 1$. Similarly, this maximization can be reformulated as a constrained optimization problem.

$$\begin{aligned} &\min_{\mathbf{a}} \ -\log\left[L(\mathbf{a} \mid \mathbf{D})\right] \\ &s.t. \quad 0 < a_i < 1 \end{aligned} \tag{2.28}$$

The obtained transformation parameters are used to model the probability density function of the normal process data. The region defined by the probability function contains features of the normal operational variations.

2.6 Online Monitoring

During online monitoring, the real-time process data samples, $\mathbf{x}^j = [x_1^j, x_2^j, \dots, x_d^j]$,are fed into Eq.(2.18) and their probabilities are computed. This probability indicates how likely one process data sample belongs to the normal region.

$$f(x_1^j, x_2^j, ..., x_d^j) = c[F(y_1^j), F(y_2^j), ..., F(y_d^j)] \prod_{i=1}^{d} f(y_i^j)(1-a_i) \tag{2.29}$$

A probabilistic control limit is then determined to define the normal region, within which all process data samples have a probability higher than the control limit. Depending upon the industrial cases studied, this control limit can be set to a specific probability; for example, 0.3%. In this respect, any real-time process data sample that has a probability less than the control limit is flagged as faulty. For fault diagnosis, the normal dependence structure between each transformed process variable and the reference variable is first determined. This dependence structure is modelled using 100 samples of the transformed normal process data by a bivariate Gaussian copula.

$$D_s^i = \frac{\sum_{j=1}^{100} c(y_i^j, x_r^j)}{100} \tag{2.30}$$

It is worthwhile to note that Y_i is strict-monotonically increasing with respect to X_r . Therefore, the bivariate Gaussian is able to accurately model the fault-free relationships between the transformed process variables and the reference variable. However, these relationships only hold for normal operation. Any fault condition that introduces undesired disturbances to the process can easily disrupt the fault-free dependence structures. To identify the root-cause process variable(s), the same bivariate Gaussian copula with the same correlation matrix in Eq. (2.30) is used to modelled the dependence structures of every process variable in relation to the reference variable using the first 100 faulty data samples after the fault is detected. These dependence structures are then compared with those of normal. The contribution of each process variable is determined by the following Equation.

$$Cont_i = \sum_{q=1}^{100} \left[c(y_i^q, x_r^q) - D_s^i \right]^2 \tag{2.31}$$

where q is faulty sample index. Finally, the complete methodology of the multivariate copula-based fault diagnosis technique is summarised in the logic flowchart, as shown in Figure 2-1.

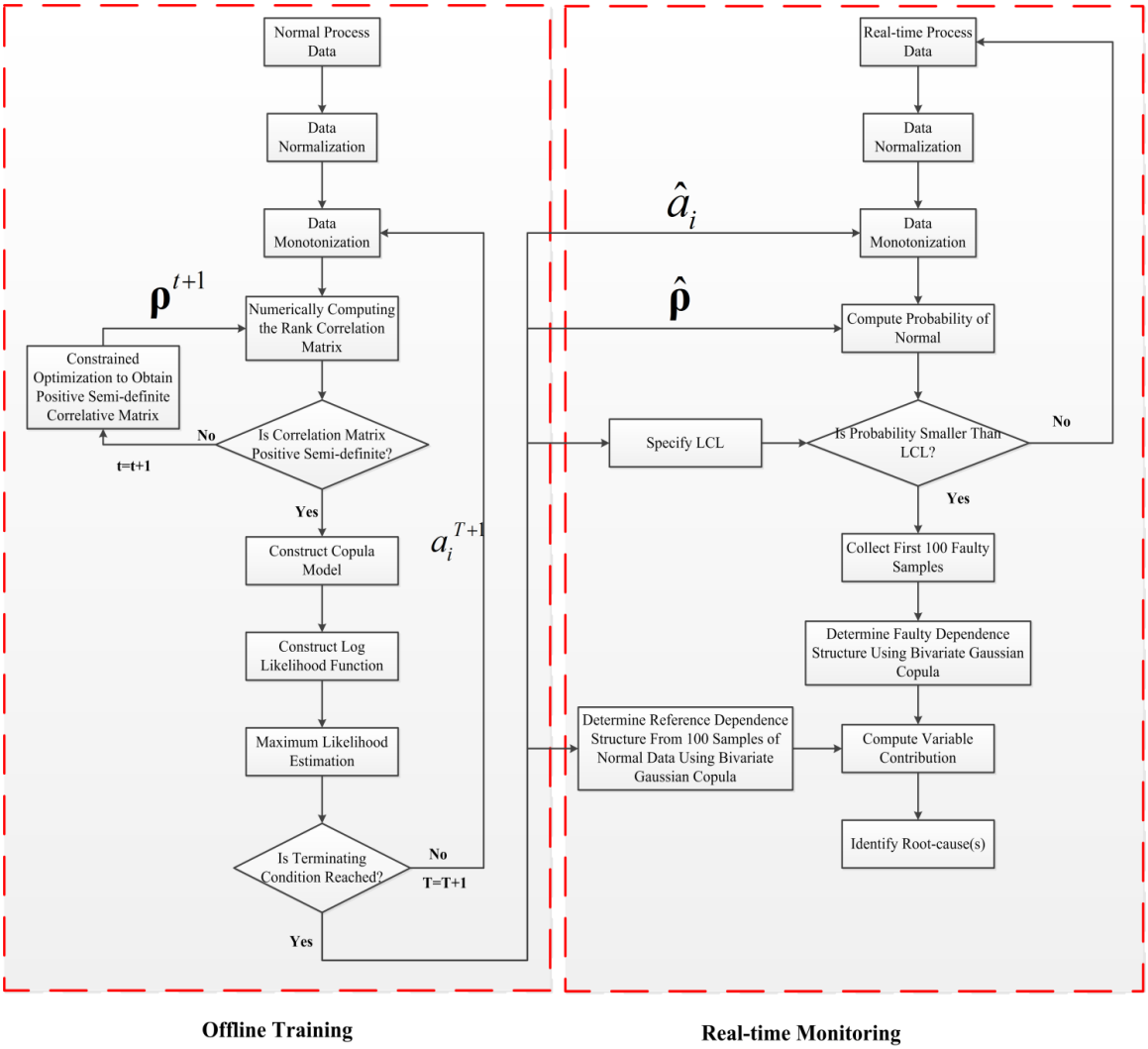

Figure 2-1: Logic flowchart of the proposed multivariate copula-based fault diagnosis technique.

2.7 Case Studies

2.7.1 Motivational example

The ability of the proposed method in discerning not linearly separable faulty samples is demonstrated in this case study. The nonlinear numerical models considered are shown as following.

$$\begin{aligned} x_1 &= -0.5 + e_1, e_1 \sim N(0,1) \\ x_2 &= x_1^2 + e_2, e_2 \sim N(0,0.02) \\ x_3 &= x_2^{-2} + e_3, e_3 \sim N(0,0.01) \\ x_4 &= x_3^2 + e_4, e_4 \sim N(0,0.005) \end{aligned} \tag{2.32}$$

It is evident that the relationship among the model variables is $x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4$. Each of them is also a nonlinear transformation of its previous model variable. It could

also be noted that the relationships within some pairs of the model variables are not monotonic; for instance, the relationship between x_1 and x_2 is quadratic. This is also shown by the rank correlation matrix of the four model variables.

$$\rho_s = \begin{bmatrix} x_1 & x_2 & x_3 & x_4 & \\ 1.00000 & -0.0050 & 0.00470 & 0.00470 & x_1 \\ -0.0050 & 1.00000 & -0.9747 & -0.9747 & x_2 \\ 0.00470 & -0.9747 & 1.00000 & 1.00000 & x_3 \\ 0.00470 & -0.9747 & 1.00000 & 1.00000 & x_4 \end{bmatrix} \tag{2.33}$$

Note that the rank correlation coefficients between x_1 and x_2, x_3 and x_4 are very close to zero, indicating the monotonic relationships within these two pairs of model variables are weak. Therefore, the copula function is not able to model the true dependence structures among these variables. On the other hand, the model for generating the faulty data samples is given as.

$$\begin{aligned} f_1 &= e_{f1}, e_{f1} \sim N(0, 0.01) \\ f_2 &= e_{f2}, e_{f2} \sim N(0.2, 0.01) \\ f_3 &= e_{f3}, e_{f3} \sim -N(0.15, 0.01) \\ f_4 &= e_{f4}, e_{f3} \sim N(0.2, 0.02) \end{aligned} \tag{2.34}$$

The distributions of 10000 data samples of the first two model variables from each model are visualized in Figure 2-2. Due to the small variances in the faulty model, the faulty data samples concentrate in a relatively small region and are not linearly separable from the normal data samples.

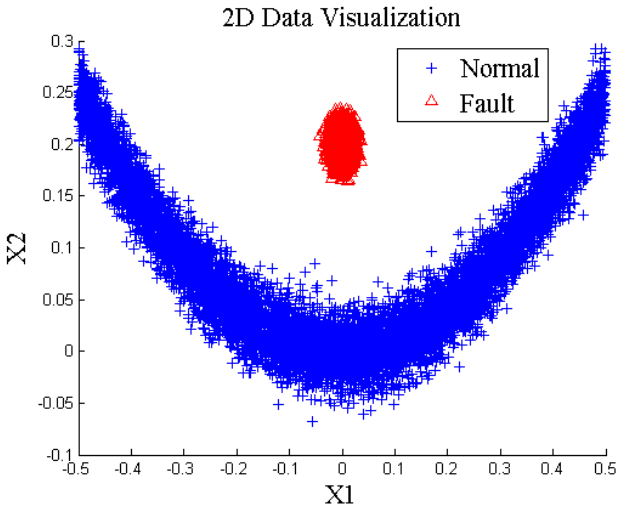

Figure 2-2: 2D visualization of data samples for the nonlinear numerical model.

The PCA and modified ICA[15] model are first used to classify the faulty data samples. Both models are trained with 1000 samples of the normal process data. The number of

principal components (PCs) and independent components (ICs) retained are determined through cross-validation. In this case study, 2PCs or ICs are retained for both methods. Figure 2-3 shows the faulty data classification results of the PCA and ICA models. In total, 2000 data samples are generated in which the first 1000 samples are normal and the remaining 1000 samples are faulty.

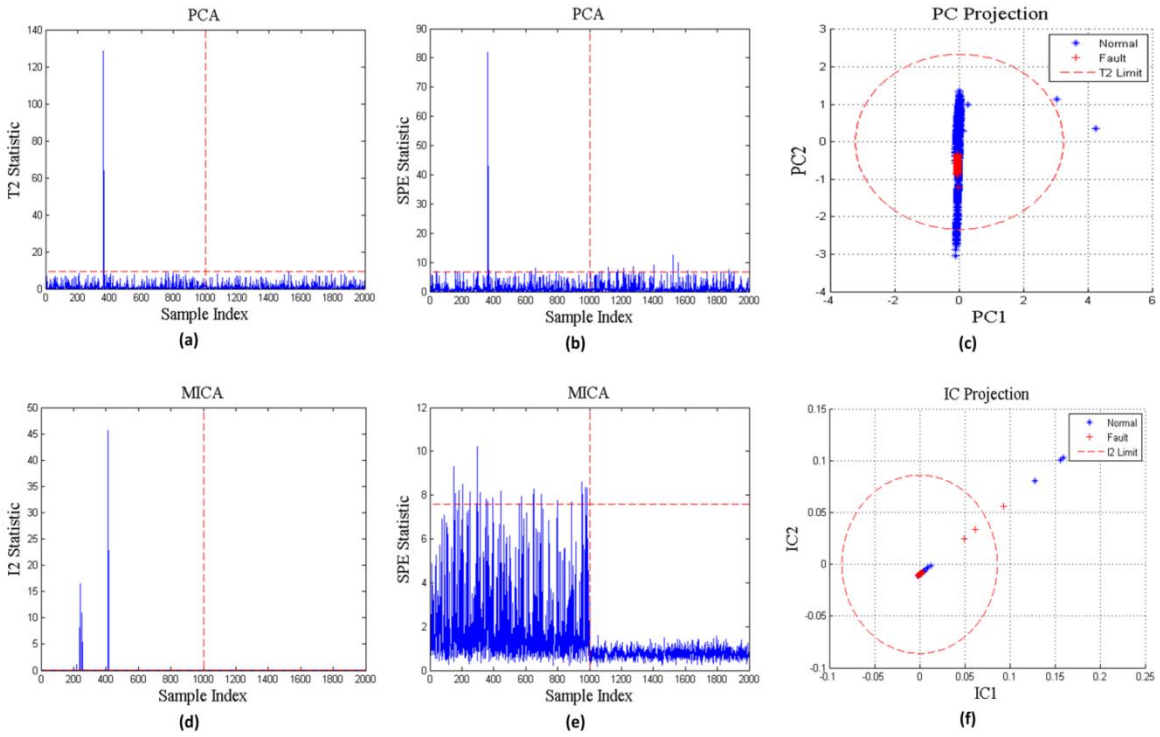

Figure 2-3: Faulty sample classification results of PCA and ICA.

Since the faulty data samples are not linearly separable from the normal samples, none of the statistics in PCA and ICA are able to classify the faulty data samples. The faulty data samples are concentrated in a relatively small region within the normal region defined by the T^2 limit and SPE limit; the SPE statistic of the MICA model has much less variation for the faulty data samples. This phenomenon could also be observed in Figure 2-3(c) and Figure 2-3(f) in which the projected faulty data samples are also concentrated in a small area well within the normal region in the PC space. On the other hand, the fault detection results of the proposed method are shown in Figure 2-4. In this case, the lower control limit is set at 0.3% implying the normal region contains approximately 99.7% of normal model variations. The values of the transformation parameters are $\mathbf{a} = [0.7587\ 0.9276\ 0.8226\ 0.8472]$. After the monotonic transformation, the rank correlation matrix is shown in Eq. (2.35). The transformed model variables have strong monotonically-increasing relationship as the rank correlation coefficients for every pair of variables are very close to 1.

$$\rho_s = \begin{bmatrix} 1 & 0.9443 & 0.9469 & 0.9474 \\ 0.9443 & 1 & 0.9957 & 0.9964 \\ 0.9469 & 0.9957 & 1 & 0.9995 \\ 0.9474 & 0.9964 & 0.9995 & 1 \end{bmatrix} \tag{2.35}$$

In addition, the comparison of the fault classification results for PCA, MICA and the proposed approach is presented in Table 2-1. The reason of high fault classification rate for the proposed technique is due to its ability to construct an accurate probability density function over the normal model data samples. Also, the nonlinear relationships between the model variables are well-captured in the copula.

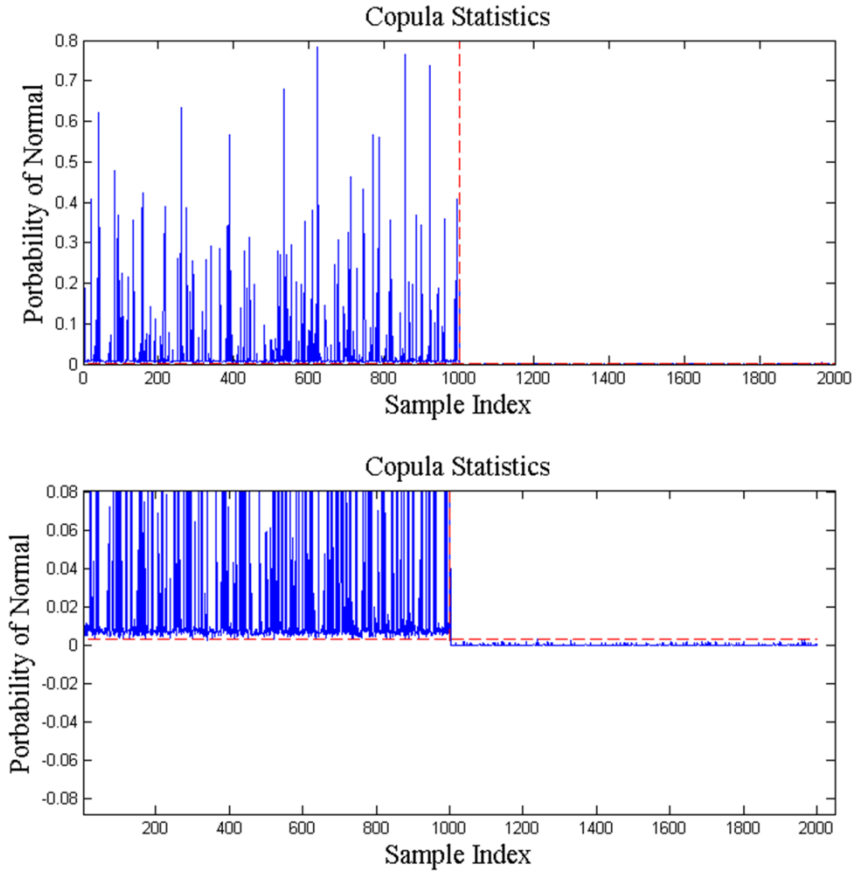

Figure 2-4: Faulty sample classification results of the multivariate copula-based technique.

Table 2-1: Comparison of fault classification results for the nonlinear numerical model

	PCA		MICA		Copula
	T^2	SPE	I^2	SPE	
FDR† [%]	0	1	0	0	99.9
FAR‡ [%]	0.2	0.9	4.3	0.6	2

† FDR: Fault Detection Rate

To further illustrate the fault classification mechanism of the proposed method, a probability density function is estimated for the 2D data in Figure 2-2 using the proposed method. In Figure 2-5, the faulty data samples are far exceeding the normal region spanned by the estimated probability density function. As a result, the probabilities of the faulty data samples are well below the lower control limit.

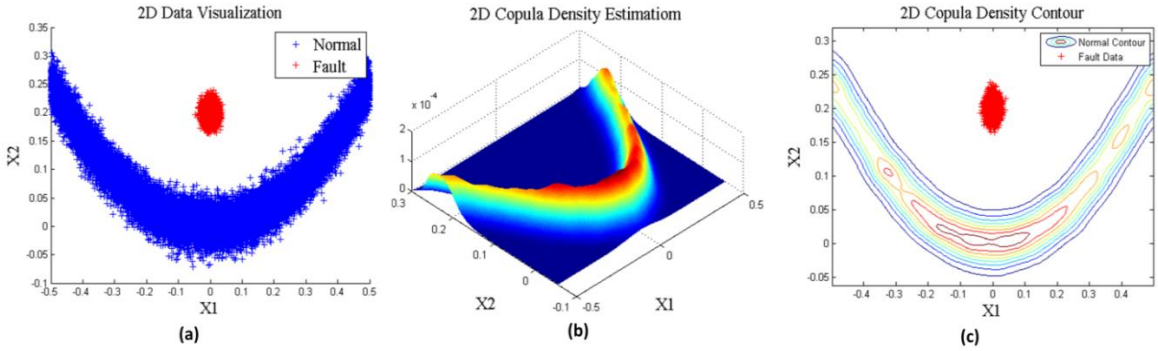

Figure 2-5: Illustrative 2D copula density estimation for fault classification.

2.7.2 Industrial case study

In this section, the effectiveness of the multivariate copula-based technique is further demonstrated using a well-studied benchmark simulation of the Tennessee Eastman chemical process. The proposed technique is used to detect a variety of fault types as well identify their root-causes. The Tennessee Eastman chemical process comprises of five major operating units: an exothermic two-phase reactor, a product condenser, a vapour-liquid flash separator, a recycle compressor, and a reboiled product stripper. The process flow diagram of the chemical plant is shown in Figure 9-1. The detailed explanation of this benchmark process can be found in the work of Downs, Vogel[32]. In total, there are 41 measured process variables in the process. In this work, 22 variables are selected to determine the operating condition of the process system. These monitored variables are listed in Table 9-1. In addition, Table 9-2 summarizes the 15 known fault conditions that are pre-programmed in the Tennessee Eastman process simulation and have been widely used by the process monitoring community.

Similar to the first case study, the PCA, MICA and the proposed technique are all applied to monitor this process. The training data for all there techniques consists of 1000 normal data samples collected at 0.05 hr sampling interval. For PCA and MICA, there are 11 PCs and ICs selected through cross-validation for real-time monitoring. It is worth noting that 22 variables are used to model the copula function and the joint probability density function of the proposed technique; the joint probability of each data sample might become extremely small due to the multiplications in Eq.(2.18) . To address this problem, the negative log value of the probability is used for online fault detection. Because of the use of negative log, the lower control limit for the probabilities is converted to the upper control limit of the negative log values. In this case, this upper control limit is set at 110 covering approximately 99% of the negative log values of the

‡ FAR: False Alarm Rate

1000 normal process data samples used for training. The fault detection results for all three techniques are summarized in Table 2-2.

Table 2-2: Fault detection rates and false alarm rates for PCA, ICA and the proposed technique.

Fault Detection Rate (%)					
Faults	**PCA [99% ULC]**		**ICA [99% UCL]**		**Copula [110 UCL]**
	T^2	*SPE*	T^2	*SPE*	*-log(p)*
1	99.76	99.86	99.79	99.76	99.81
2	99.43	99.60	99.43	99.38	99.75
3	2.09	1.56	2.14	2.71	***22.52****
4	0.45	1.07	2.24	2.31	***19.75****
5	1.43	1.19	1.59	1.74	***10.33****
6	99.86	100	100	100	100
7	2.45	4.18	10.26	6.81	**22.95***
8	91.32	94.36	98	97.6	99.82
9	1.09	0.93	6.78	3.31	***21.38****
10	1.59	40.62	71.24	75.15	***86.96****
11	27.80	51.24	71.86	86.31	83.12
12	34.85	21.35	41.01	27.69	***51.44****
13	84.10	93.03	93.79	93.62	96.88
14	25.56	99.67	94.26	99.6	91.10
15	1.12	1.21	2.95	2.05	***11.44****
False Alarm Rate (%) under 99% Confidence UCL					
	T^2	*SPE*	T^2	*SPE*	*-log(p)*
	1.37	0.7	2.67	1.63	2.23

The fault conditions in which the proposed method outperforms the PCA and MICA are marked in bold. In these fault conditions, PCA has performed the worst in terms of the fault detection rates. In particular, for fault condition 4, the fault detection rate is less than 1%. This is mainly due to the assumption of ideal process operation in which variable interactions are linear, the generated process data is noise free and the latent variables governing the latent features of the operation follow ideal Gaussian distribution. With regard to MICA, the ability to capture non-Gaussian variations in the latent space has resulted in significant improvement in performance over PCA. Especially, for fault conditions 7, 10, 11 and 12, the performance is improved by more than 20% on average. Nevertheless, MICA does not address the problem of process noise and nonlinearity, thus leading to suboptimal performance. In contrast, the proposed technique performs better in almost all the fault conditions though the fault detection rates for some of the difficult cases are still below 30%. These low fault detection rates are caused by the decentralized closed-loop stable control strategy implemented in the simulation program which quickly corrects the undesired disturbances. In spite of this, for the other fault conditions which cannot be corrected by the control actions, the proposed technique shows promising performance. To further demonstrate the fault detection performance of all three techniques, the process monitoring charts of PCA, MICA and the proposed technique for

fault conditions 10 and 13 are shown in Figure 2-6 and Figure 2-7. All fault conditions are introduced at sample index 3000.

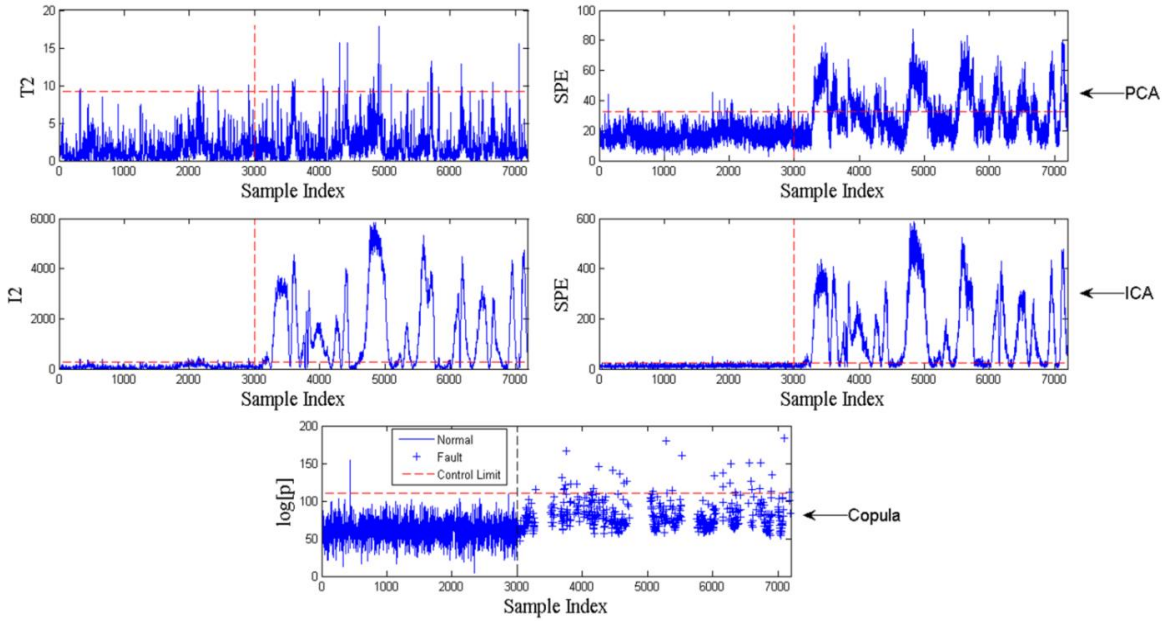

Figure 2-6: Process monitoring charts for IDV10 based based on PCA, MICA and Copula-based method

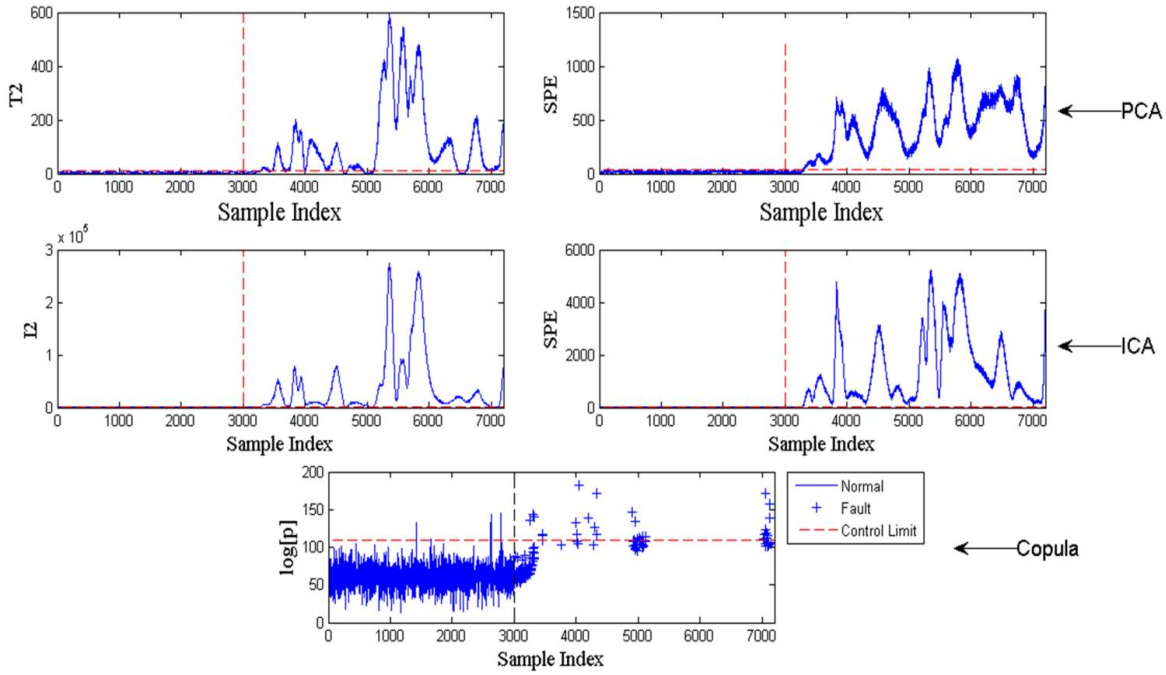

Figure 2-7: Process monitoring charts for IDV13 based on PCA, MICA and Copula-based method.

It is noticed that for the proposed method, most of the samples disappear after the fault is introduced at sample index 3000. This is due to the fact that these samples far exceed the normal region and their probabilities of normal reach nearly zero. When converting probability into log domain, a nearly zero value gives rise to a significant underflow

issue, resulting in an infinite large value. These values stretch out of the tolerance bound of any computing language and are therefore not displayed in the charts.

For fault diagnosis, two fault conditions have been selected for testing, which are IDV6 and IDV10. The first 100 faulty data samples after the fault is detected are used to compute variable contributions. The fault diagnosis results of the PCA, MICA and the proposed method for both tested conditions are presented in Figure 2-8 and Figure 2-9, respectively. In IDV6, a step decrease in the feed A is introduced into the process. The first process variable that is impacted by this fault condition is the A feed (X1). As shown in Figure 2-8, the abnormal behaviour of this variable is correctly captured by the proposed method. Due to the loss in feed A, the reactor feed rate is also adversely affected. Subsequently, the dynamic balance of chemical reaction in the reactor tank is disrupted, resulting in undesired behaviour of the reactor temperature (X9). This radical change in temperature is also reflected in the reactor cooling water outlet temperature (X21) which is then correctly identified by the PCA-based fault diagnosis. The SPE statistic of the MICA also correctly captures the root-cause variable; however, the other unrelated process variables also show high contribution to the fault.

For IDV10, the temperature of C feed is not monitored. The immediate downstream variable related to C feed temperature is the Stripper temperature (X18). The proposed method is able to correctly identify this most closely related process variable. As shown in Figure 2-9, the stripper temperature (X18) has the highest contribution. Also, C feed participates in the stripping of the condensed product stream from the separator to remove residual reactants. The residual reactants are then recycled back to reactor through stream 5 as shown in Figure 9-1. This flow of product streams could serve as a propagation path for IDV10. As a result, the recycle flow (X5) also shows abnormal behaviour and eventually causes problem in reactor temperature (X9). In fact, this fault propagation path is also correctly recognized by the proposed method. Meanwhile, the I^2 statistic of the MICA also provides the correct diagnosis, but the correct fault propagation path is not identified. On the other hand, in the PCA contribution charts, several non-related process variables show high contribution to the fault.

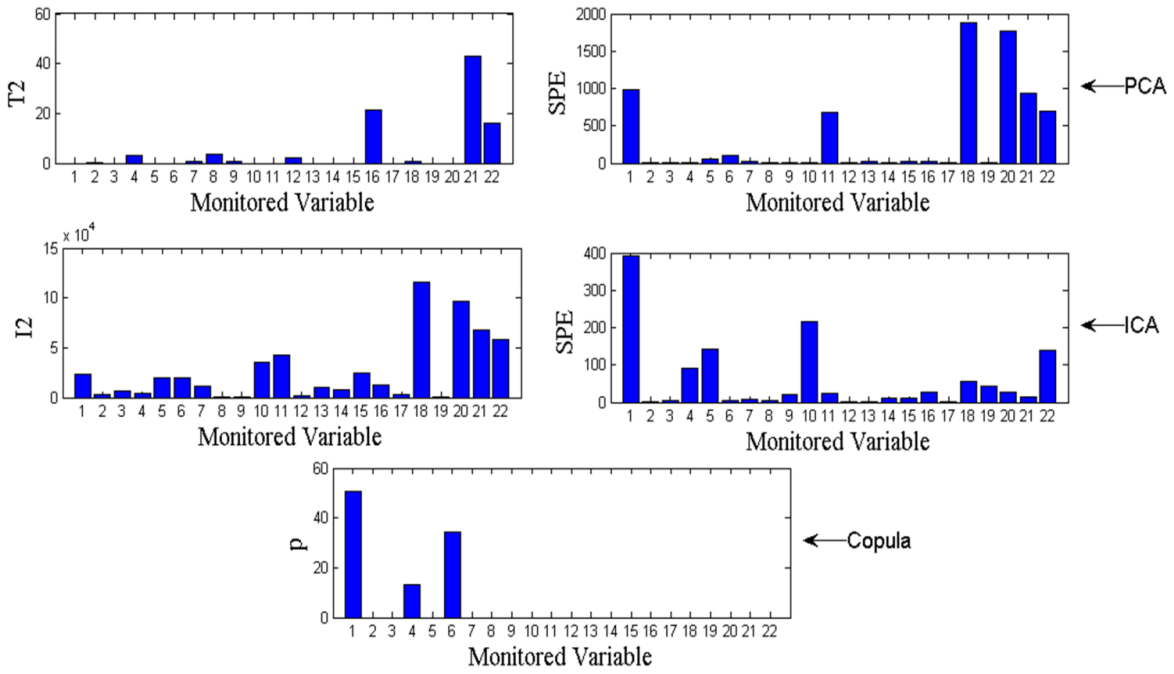

Figure 2-8: Comparison of the fault diagnosis results of PCA, MICA and the Copula-based method for IDV6

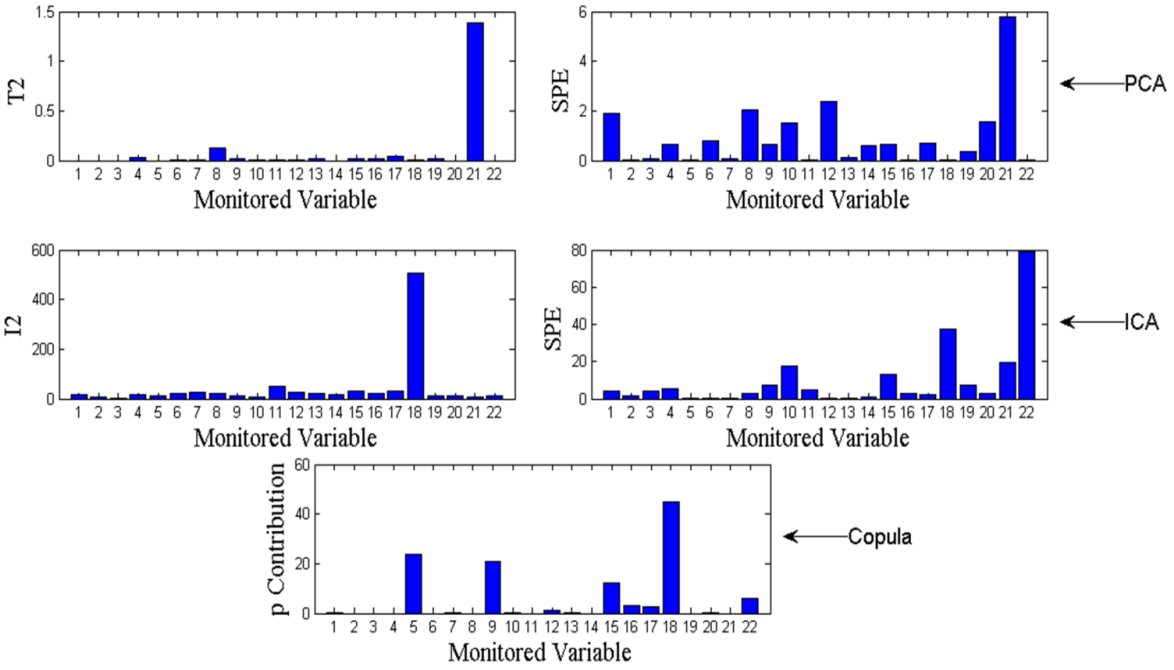

Figure 2-9: Comparison of the fault diagnosis results of PCA, MICA and the Copula-based method for IDV10.

2.8 Conclusion

This article presents a multivariate probabilistic process monitoring technique for industrial processes. The proposed technique constructs an accurate joint probability density function using copula to define the normal regime of process operation. In fact, the joint probability density function is modelled using two independent components: the

copula density function and the univariate marginal distributions of the process variables. The copula density function captures the complete dependence structures among the process variables; however it requires pairwise relationships between the process variables to be strict-monotonically increasing. The Rolling Pin method is adopted to monotonize the pairwise relationships, which uses Maximum Likelihood Estimation to determine a set of transformation parameters that not only ensure strong monotonicity but also enough information regarding process variation is preserved in the transformed process variables. On the other hand, the random variables in the copula density function are the univariate marginal distributions of the transformed process variables. These univariate marginal distributions are estimated using non-parametric kernel estimator. The correlation matrix for the copula density function is computed numerically. A numerical optimization method is also proposed to condition the correlation matrix so that it is always positive semi-definite. In essence, this correlation matrix contains the rank correlation coefficients of the transformed variables. This allows copula to model extremely nonlinear relationships in complex processes.

During real-time monitoring, the probability of normal of each process sample is computed under the obtained joint probability density function. Those process samples having probabilities less than a predefined lower control limit are classified as faulty samples. For fault diagnosis, the reference structures between each transformed variable and the reference variable are first modelled using a bivariate Gaussian copula. These reference structures are then compared with the structures obtained from the first 100 faulty data samples. Finally, the root-cause(s) are identified to be the process variable(s) deviating the most from their reference structures. The proposed technique has been tested using two case studies: a motivational example and the benchmark Tennessee Eastman Process. The proposed technique demonstrated superior performance to the conventional statistical techniques, such as PCA and MICA.

3 A Sparse PCA for Non-linear Fault Diagnosis and Robust Feature Discovery Industrial Processes

Abstract

Pearson's correlation measure is only able to model linear dependence between random variables. Hence, conventional principal component analysis (PCA) based on Pearson's correlation measure is not suitable for application to modern industrial processes where process variables are often nonlinearly related. To address this problem, a non-parametric PCA model is proposed based on nonlinear correlation measures, including Spearman's and Kendall tau's rank correlation. These two correlation measures are also less sensitive to outliers comparing to Pearson's correlation, making the proposed PCA a robust feature extraction technique. To reveal meaningful patterns from process data, a generalized iterative deflation method is applied to the robust correlation matrix of the process data to sequentially extract a set of leading sparse pseudo-eigenvectors. For online fault diagnosis, the T^2 and SPE statistics are computed and analysed with respect to the subspace spanned by the extracted pseudo-eigenvectors. The proposed method is applied to two industrial case studies. Its process monitoring performance is demonstrated to be superior to that of the conventional PCA and is comparable to those of Kernel PCA and kernel independent component analysis (KICA) at a lower computational cost. The proposed PCA is also more robust in sparse feature extraction from contaminated process data.

Keywords: Principal Component Analysis, Spearman's rank correlation, Kendall tau's rank correlation, nonlinear process monitoring, robust feature discovery.

3.1 Introduction

Modern industrial processes comprise of a large number of non-linear subsystems. These subsystems also interact with each other in a complex fashion. In most cases, it is difficult to obtain an explicit model to accurately describe the dynamical behaviour of the systems. Due to the absence of such a model, the use of traditional first-principle model-based process monitoring techniques suffers great limitations. This has led to extensive application of statistical data-driven process monitoring techniques.[2,9,33] In general, these techniques determine a normal subspace spanned by a set of vectors pointing towards the directions of most variation of the normal data. It is often assumed that the normal subspace holds critical information regarding the stochastic behaviour of normal operating processes. On the other hand, the unexplained variations are collected in a residual space. In real-time process monitoring, on-line process data samples are projected into both subspaces. The "portion" of the on-line data samples in each subspace is measured for anomaly detection. Subsequently, further decomposition of the projected data samples reveals the highest contributing process variable for root-cause identification.

Multivariate data analysis is the core of the statistical data-driven process monitoring technique. Each data sample is considered to be drawn from a multivariate probability distribution. The aim of the multivariate data analysis is to determine the best fit probability distribution--a probability distribution that maximizes the likelihood of the data samples.[34,35] Depending upon the initial assumption of the data distribution, the determined probability distribution can be parametric or non-parametric. When the data samples are assumed to follow a multivariate Gaussian distribution, eigenvalue decomposition can be applied to the covariance matrix to determine a set of orthonormal vectors known as eigenvectors. Each eigenvector represents the major axis of an independent Gaussian component whose variance is the associated eigenvalue. The multivariate probability distribution of the data samples is the joint distribution of the independent Gaussian components. This particular type of multivariate data analysis is known as the Principal Component Analysis or PCA.[36]

There are four major drawbacks associated with the standard PCA when applied to monitor complex processes: (1) the covariance matrix is scale variant making it difficult to accurately model dependence structures among process variables with different measurement scales;[37] (2) the covariance matrix is obtained by using the Pearson's dependence measure, which is sensitive to outliers incapacitating the robustness of the standard PCA;[38,39] (3) the Pearson's dependence measure is only able to capture linear dependence between random variables. As a result, nonlinear features of the process data cannot be retained in the standard PCA model; (4) when the process data samples do not follow Gaussian distribution, the extracted eigenvectors are not able to explain the majority of the process data variance. It is noticed that the main source of the weaknesses of the standard PCA model is the use of Pearson's dependence measure. The first drawback can be easily counteracted by standardizing the process data samples. The covariance matrix of the standardized data samples becomes the correlation matrix of the original data samples, which is scale invariant. Subsequently, eigenvectors are extracted

from the correlation matrix. This special treatment of the standard PCA is referred to as the scale-invariant PCA.[40] In fact, almost all applications of PCA in process monitoring adopt this scale-invariant feature extraction procedure. When considering the second drawback, the covariance is in essence the second moment of the data. An outlier data sample deviating considerably from the mean (centre) of the data distribution causes significant distortion of the covariance matrix. This distortion may induce the extraction of unnecessary eigenvectors in the direction of outlier data samples to account for the additional variance.

To explore non-linear features of the process data samples, the Kernel extensions of PCA were developed.[18,20] These two techniques rely on kernel mapping which is formulated based on Vapnik-Chervonenkis theory—data samples that are not classifiable in low dimensional space become linearly separable in a much higher dimensional space. However, kernel mapping can be computationally expensive if there are a lot more data samples than features. In addition, kernel mapping is irreversible, meaning that the identification of root-cause through reverse projection is practically impossible. Furthermore, after the kernel mapping, the same PCA method is applied to the mapped data to extract latent features. This implies that the non-robustness of standard PCA is also inherited.

A number of alternative methods have been proposed to address the fourth major drawback of the standard PCA. Independent component analysis or ICA is one of the most widely applied non-Gaussian feature extraction techniques.[41] In contrast to PCA, ICA determines a set of non-orthogonal unit vectors that are as independent as possible.[16] ICA has been demonstrated to outperform PCA on data with dominating non-Gaussian features.[1,42-44] However, ICA still suffers from the second major drawback of the PCA, primarily due to the usage of PCA in the data whitening step and the adoption of kurtosis (involving third and fourth moment of data samples) to measure non-Gaussianity. The same problem also exists with the Kernel ICA. Furthermore, the advantage of ICA in explaining non-Gaussian variance becomes less significant when dealing with high dimensional data. Essentially, the sub-space components of both PCA (PC components) and ICA (IC components) are obtained by weighted linear combinations of the original process variables. According to the central limit theorem, the sum of a large number of i.i.d. random variables is prone to be Gaussian distributed.[45] In this regard, the variance of the sub-space components of high dimensional data can still be well explained by eigenvectors extracted using conventional PCA.

In this work, a robust PCA model is proposed to address the aforementioned major drawbacks of the standard PCA. First, industrial process data samples are robustly standardized using their median and median absolute deviation (MAD).[46] Subsequently, the Spearman's and Kendall tau's rank correlation coefficient is computed for each pair of process variables to form the rank correlation matrix. The Spearman's and Kendall tau's rank correlation coefficients are robust to outliers. They scale down the large magnitude of deviation of the outlier data sample to its rank. Therefore, the distortion from the outlier to the correlation matrix is significantly reduced.[47] Finally, a set of eigenvectors retaining the nonlinear correlation structures of the process data are extracted from the rank correlation matrix to construct the feature space. To make this

technique more appealing, a generalized iterative deflation procedure[48] is applied to the rank correlation matrix to extract sparse eigenvectors. The sparsity in the eigenvectors has a number of desirable features. It can reduce the computational effort of subspace projection, particularly for large-scale processes. Additionally, it can reveal more meaningful patterns from data and has better generalization capability.[48-50] After the sparse eigenvectors are extracted, on-line process data samples are projected into the subspace. The T^2 and SPE statistics are computed for fault detection and root-cause identification.

The remainder of the manuscript is divided as follows. Section 2 introduces the set of mathematical notations adopted, followed by a brief introduction of the basic techniques necessary for formulating the proposed PCA model. The complete methodology of the proposed sparse PCA is then explained in detail in section 3. In section 4, two case studies including a numerical example and an industrial case study are used to evaluate the performance of the proposed technique. A comparison of performances among PCA, KPCA, KICA, and the proposed technique is also presented in section 4 to further demonstrate the strength of the proposed sparse PCA. Finally, the major findings and conclusion of this research are summarized in section 5.

3.2 Preliminaries

Let $\mathbf{X} = \{X_1, X_2, \dots, X_d\}$ denote a set of d continuous random variables. A single data sample of $\mathbf{X}$ is expressed as $\mathbf{x}^i = \{x_1^i, x_2^i, \dots, x_d^i\}, \mathbf{x}^i \in \mathbb{R}^d$. The small letter x_j^i, $\forall i \in \{1,2,\dots,N\}$, $\forall j \in \{1,2,\dots,d\}$, represents the numerical value that variable j can take at sampling interval i, where N is the total number of samples. Three types of vector norms adopted in this article are: l_0, l_1, l_2, where $l_0 = ||\mathbf{x}||_0 = \text{card}(\text{supp}(\mathbf{x}))$, $l_1 = ||\mathbf{x}||_1 = \sum_{j=1}^{d} |x_j|$ and $l_2 = ||\mathbf{x}||_2 = \sum_{j=1}^{d} |x_j|^2$. The covariance matrix and the correlation matrix are given the Greek letters $\boldsymbol{\Sigma}$ and $\boldsymbol{\Sigma^0}$ respectively. The eigenvectors $\{\mathbf{v}_j\}_{j=1}^{d}$ of $\boldsymbol{\Sigma^0}$ are sorted according to the descending order of their respective eigenvalues. For instance, the eigenvalue λ_1 corresponding to $\mathbf{v}_1$ is the largest eigenvalue; conversely, λ_d associated with $\mathbf{v}_d$ is the smallest eigenvalue. The inner product of vectors $\mathbf{a}$ and $\mathbf{b}$ is represented as $\langle \mathbf{a}, \mathbf{b} \rangle$. Finally, let $|\mathbf{X}|$ be an operator that takes the absolute value of each element of $\mathbf{X}$.

3.2.1 Spearman's and Kendall tau's Rank Correlation

Both Spearman's and Kendall tau's rank coefficient provide robust measure of statistical dependence between random variables.[51] For Spearman's rank correlation, the ranks of the numerical values of each random variable are used to calculate the variance and covariance. Let s_j^i be the rank of the i^{th} numerical value of the j^{th} random variable. Similarly, let s_k^i denote the rank of the k^{th} random variable. The Spearman's rank correlation between random variables j and k is calculated as:

$$\hat{\sigma}_s\langle jk\rangle = \frac{\sum_{i=1}^{N}(s_j^i - \hat{s}_j)(s_k^i - \hat{s}_k)}{\sqrt{\sum_{i=1}^{N}(s_j^i - \hat{s}_j)^2 \cdot \sum_{i=1}^{N}(s_k^i - \hat{s}_k)^2}}. \quad (3.1)$$

where, $\hat{s}_{j=}\hat{s}_k = \frac{N+1}{2}$. On the other hand, the Kendall tau's rank correlation coefficient is computed based on the number of concordant and discordant pairs of samples between two random variables. Specifically, for two random variables X and Y, any pair of samples $\langle x_a, y_a\rangle$ and $\langle x_b, y_b\rangle$ are considered to be concordant if $x_a > x_b, y_a > y_b$ or $(x_a < x_b, y_a < y_b)$; the violation of these conditions will render the pair of samples discordant with an exception of $x_a = x_b, y_a = y_b$, which is said to be neither concordant or discordant. Then, the Kendall tau's rank correlation coefficient is computed as:

$$\hat{\rho}_s\langle jk\rangle = \begin{cases} \dfrac{N_c - N_d}{1/2\, N(N-1)} & \text{if } j \neq k; \\ 1 & \text{if } j = k. \end{cases} \quad (3.2)$$

where N_c and N_d denotes the number of concordant and discordant pairs of samples respectively. In addition to their robustness, both of these two correlation coefficients have better performances as compared to the Pearson's correlation coefficient in terms of capturing non-linear relationship.[52,53] This is demonstrated in the following four examples.

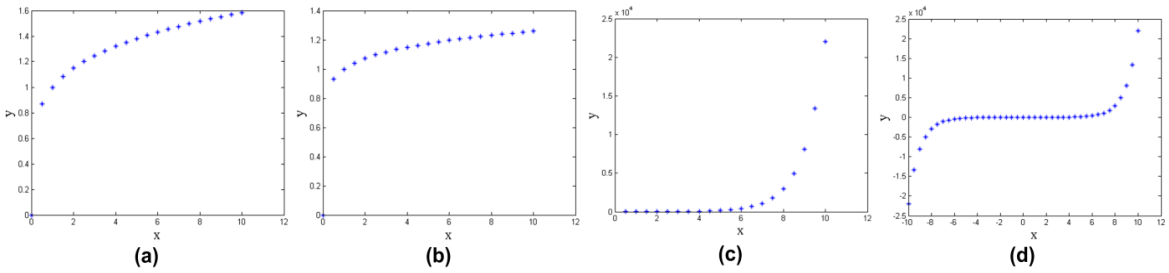

Figure 3-1: Non-linear relationship examples for Pearson's, Spearman's and Kendall tau's correlation coefficient.

Four types of monotonic non-linear relationships are shown in Figure 3-1. It is observed that there exits strong relationship between random variables x and y in all of these examples. The strength of correlation obtained from the Pearson's, Spearman's and Kendall tau's coefficient is summarized in Table 3-1.

Table 3-1: Correlation coefficients for the non-linear example

	(a)	(b)	(c)	(d)
Pearson's Coefficient	0.6962	0.699	0.6374	0.8014
Spearman's Coefficient	1	1	1	1
Kendall tau's Coefficient	1	1	1	1

Although there is a strong relationship between X and Y in all the examples, it is evident that the Pearson's coefficient is only able to describe part of the relationship. In contrast, both Spearman's coefficient and Kendall tau's Coefficient are capable of fully capturing the dependence structure. This feature of Spearman's and Kendall tau's rank coefficient allows more information of the nonlinear process variations to be retained in the correlation matrix, thus leading to better process monitoring performance. Regarding the robustness, both Spearman's and Kendall tau's correlation measures have been verified in the work of Croux, Dehon[51], with Kendall tau's correlation having better performance in terms of asymptotic efficiency, gross-error sensitivity, and lower Mean Squared Error for correlation coefficient estimation under high data contamination rate. In this study, the performance of these two robust correlation measures will be compared through an industrial case study.

3.2.2 Sequential eigenvector extraction

The sequential eigenvector extraction method is proposed as an alternative method to the standard PCA.[54] It is an iterative method consisting of two main procedures. In the first step, the first leading eigenvector is extracted from the correlation matrix. Then, the variance associated with this leading vector is removed from the old correlation matrix to form a new correlation matrix. These two steps are then reiterated until a predefined number of leading eigenvectors are extracted. The concept of sequential eigenvector extraction method is based on the following three propositions.

Proposition 1. *The first leading vector v_1 of a correlation matrix Σ^0 represents the direction of the largest variance. The amount of variance in the direction of v_1 is equal to the associated eigenvalue λ_1.*[55]

The proof of Proposition 1 is shown in Section 9.3 Appendices. The next step is to remove only the variance associated with the first leading eigenvector. This is achieved by deflating the covariance matrix with the first leading eigenvector. The Hotelling's deflation method is one of the simplest techniques, which takes the form in Eq. (9.2). For $t \in \mathbb{Z}^+, \sup \mathbb{Z}^+ = q$, where q is the desired number of eigenvectors to be extracted, $\mathbf{\Sigma}_{t=0}^0 = \mathbf{\Sigma}^0$ and $\mathbf{v}_t$ is the first leading eigenvector of $\mathbf{\Sigma}_{t-1}^0$. According to Proposition 1, $\mathbf{v}_t = \underset{\mathbf{v}}{\operatorname{argmax}}\, \mathbf{v}^T \mathbf{\Sigma}_{t-1}^0 \mathbf{v}$.

$$\mathbf{\Sigma}_t^0 = \mathbf{\Sigma}_{t-1}^0 - \mathbf{v}_t \mathbf{v}_t^T \mathbf{\Sigma}_{t-1}^0 \mathbf{v}_t \mathbf{v}_t^T. \tag{3.3}$$

Proposition 2. *Only the variance associated with $\mathbf{v}_t$ at step t is removed from $\mathbf{\Sigma}_{t-1}^0$ through Hotelling's matrix deflation.*[48]

The proof of Proposition 2 is shown in Section 9.5 Appendices. In fact, the first leading eigenvector $\mathbf{v}_t$ at step t of the Hotelling's deflation method is the t^{th} leading vector of the initial correlation matrix $\mathbf{\Sigma}^0$. This claim is proven in the following proposition.

Proposition 3. *At step t of the Hotelling's deflation method,* $\boldsymbol{v}_t$ *is the* t^{th} *leading eigenvector of* $\boldsymbol{\Sigma}_{t=0}^{0} = \boldsymbol{\Sigma}^{0}$.[54]

The proof of Proposition 3 is shown in Section 9.6 Appendices. The sequential eigenvector extraction method is the foundation of the sequential sparse PCA.

3.3 Methodology

The detailed steps of formulating and implementing the proposed sparse PCA are illustrated in this section. First, the process data samples are robustly centred and scaled using their medians and median absolute deviations (MAD). Correlation matrices formed by computing the Spearman's or Kendall tau's rank correlation coefficients between each pair of process variables are called the rank correlation matrices. Then, the sequential eigenvector extraction method is adapted to extract sparse eigenvectors from the rank correlation matrix of the processed data. In the last subsection, the online fault diagnosis of industrial processes based on the robust sparse PCA is formulated. For a more logic presentation of the methodology, a flow diagram explaining each crucial step of the proposed PCA technique is shown in Figure 3-2.

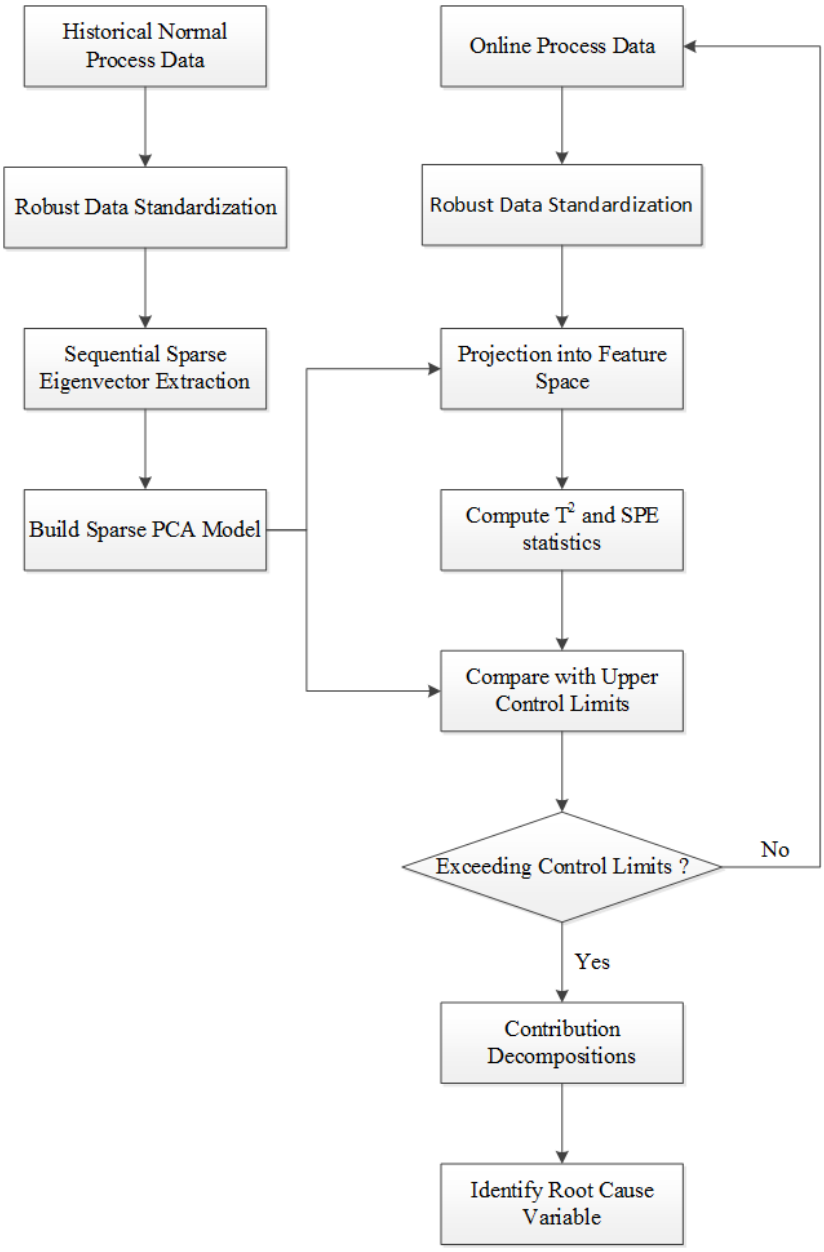

Figure 3-2: Logic flow diagram of the proposed Semi-parametric PCA.

3.3.1 Sequential sparse PCA

The eigenvectors extracted from the standard scale-invariant PCA are not sparse. There are a number of advantages of having sparsity in eigenvectors. For instance, when monitoring large-scale complex systems, sparse eigenvectors provide a good presentation of latent features. However, due to the constraint on a certain level of sparsity, the extracted sparse eigenvectors are not the real eigenvectors of a correlation matrix—they are not orthogonal to each other, meaning that the standard eigenvalue decomposition cannot be applied to extract them. A number of authors have adopted the Hotelling's iteration method directly to extract sparse eigenvectors.[49,56-58] This treatment of the sparse eigenvector extraction problem is not justified, as the crucial condition of the Hotelling's deflation method is compromised—the sparse eigenvectors are not orthonormal. To demonstrate this violation, the sparse eigenvector extraction problem is first formulated.

Suppose $\mathbf{s} \in \text{span}\{\mathbf{X}\}$ is a random unit vector and has a minimum sparsity $k, \text{card}(\text{supp}(\mathbf{s})) \leq k$. The maximum variation direction of $\mathbf{s}$ is obtained through the following optimization problem, which has an additional sparsity constraint as compared to that of the sequential eigenvector extraction problem in Section 3.2.2..

$$\hat{\mathbf{s}} = \underset{||\mathbf{s}||_2=1, ||\mathbf{s}||_0 \leq k}{\arg\max}\, \mathbf{s}^T \boldsymbol{\Sigma}^0 \mathbf{s} \tag{3.4}$$

This optimization problem is NP-hard due to the l_0 constraint on $\mathbf{s}$.[59] A convex formulation is proposed by d'Aspremont, et al.[56] to relax the l_0 constraint.

$$\begin{aligned} &\max \quad \text{Tr}(\boldsymbol{\Sigma}^0 \mathbf{s}\mathbf{s}^T) \\ &s.t. \quad \text{Tr}(\mathbf{s}\mathbf{s}^T) = 1 \quad \mathbf{1}^T |\mathbf{s}\mathbf{s}^T| \mathbf{1} \leq k \quad \mathbf{s}\mathbf{s}^T \succ 0 \end{aligned} \tag{3.5}$$

Eq. (3.5) can be effectively solved with any semi-definite programming package, such as the CVX package.[60] Alternatively, the L_0 norm on $\mathbf{s}$ can be relaxed using a convex L_1 norm,[61] through which Eq. (3.4) is reformulated to be.

$$\hat{\mathbf{s}} = \underset{||\mathbf{s}||_2=1, ||\mathbf{s}||_1 \leq \beta}{\arg\max}\, \mathbf{s}^T \boldsymbol{\Sigma}^0 \mathbf{s} \tag{3.6}$$

β is adjusted such that $||\mathbf{s}||_0 \leq k$. Eq. (3.6) can be solved using fmincon package of Matlab. Now, let $\hat{\mathbf{s}}$ denote the solution of Eq.(3.5), which is a pseudo eigenvector of $\boldsymbol{\Sigma}^0$. Then, the variance associated with $\hat{\mathbf{s}}$ is removed from $\boldsymbol{\Sigma}^0$ using the Hotelling's deflation method.

$$\hat{\boldsymbol{\Sigma}}^0 = \boldsymbol{\Sigma}^0 - \hat{\mathbf{s}}\hat{\mathbf{s}}^T \Sigma^0 \hat{\mathbf{s}}\hat{\mathbf{s}}^T \tag{3.7}$$

It is shown in the next lemma that Eq.(3.7) leads to the removal of excessive variance from correlation matrix $\boldsymbol{\Sigma}^0$.

Lemma 1. *Deflating a correlation matrix $\boldsymbol{\Sigma}^0$ with a pseudo eigenvector $\hat{\mathbf{s}}$ that is not orthogonal to every true eigenvectors of $\boldsymbol{\Sigma}^0$ leads to the removal of excessive variance.*

Proof. Suppose $\mathbf{u}$ and $\mathbf{w}$ are both true eigenvectors of $\boldsymbol{\Sigma}^0$, $\mathbf{u}^T\mathbf{w} = 0, \hat{\mathbf{s}}^T\mathbf{w} \neq 0$ and $\hat{\mathbf{s}}^T\mathbf{u} \neq 0$. $\boldsymbol{\Sigma}^0$ is first deflated by $\mathbf{u}$. According to Proposition 3, the following equality holds.

$$\sigma_{\mathbf{w}}^2 = \mathbf{w}^T \left(\boldsymbol{\Sigma}^0 - \mathbf{u}\mathbf{u}^T \boldsymbol{\Sigma}^0 \mathbf{u}\mathbf{u}^T \right) \mathbf{w} = \lambda_w. \tag{3.8}$$

where λ_w is the eigenvalue associated with $\mathbf{w}$. Then $\boldsymbol{\Sigma}^0$ is deflated by $\hat{\mathbf{s}}$.

$$\begin{aligned}\hat{\sigma}_{\mathbf{w}}^2 &= \mathbf{w}^T\left(\boldsymbol{\Sigma}^0 - \hat{\mathbf{s}}\hat{\mathbf{s}}^T\boldsymbol{\Sigma}^0\hat{\mathbf{s}}\hat{\mathbf{s}}^T\right)\mathbf{w} \\ &= \mathbf{w}^T\boldsymbol{\Sigma}^0\mathbf{w} - \mathbf{w}^T\hat{\mathbf{s}}\hat{\mathbf{s}}^T\boldsymbol{\Sigma}^0\hat{\mathbf{s}}\hat{\mathbf{s}}^T\mathbf{w} \\ &= \mathbf{w}^T\boldsymbol{\Sigma}^0\mathbf{w} - \langle\mathbf{w},\hat{\mathbf{s}}\rangle\hat{\mathbf{s}}^T\boldsymbol{\Sigma}^0\hat{\mathbf{s}}\langle\mathbf{w}_1,\hat{\mathbf{s}}\rangle \\ &= \mathbf{w}^T\boldsymbol{\Sigma}^0\mathbf{w} - \hat{\mathbf{s}}^T\boldsymbol{\Sigma}^0\hat{\mathbf{s}}\cos^2\hat{\theta}\end{aligned} \tag{3.9}$$

where $\hat{\theta}$ is the angle between $\hat{\mathbf{s}}$ and $\mathbf{w}$. Since $\hat{\mathbf{s}}$ is not orthogonal to $\mathbf{w}$, $\cos\theta \neq 0$ and $\cos^2\hat{\theta} > 0$. In addition, $\boldsymbol{\Sigma}^0$ is positive semi-definite. Therefore $\hat{\mathbf{s}}^T\boldsymbol{\Sigma}^0\hat{\mathbf{s}} \geq 0$. Let $\hat{\mathbf{s}}^T\boldsymbol{\Sigma}^0\hat{\mathbf{s}}\cos^2\hat{\theta} = C > 0$, Eq. (3.9) is modified as:

$$\hat{\sigma}_{\mathbf{w}}^2 = \mathbf{w}^T\boldsymbol{\Sigma}^0\mathbf{w} - C \leq \lambda_w \tag{3.10}$$

As $\hat{\sigma}_{\mathbf{w}} \leq \sigma_w$, it is proven that deflating $\boldsymbol{\Sigma}^0$ with $\hat{\mathbf{s}}$ leads to the removal of excessive variance.

In the subsequent step of the Hotelling's deflation method, a new pseudo eigenvector parallel to previously extracted pseudo eigenvectors could be obtained to account for the excessive variance removed. According to Eq.(3.10), for a large enough C, the eigenvalue corresponding to the true eigenvector w could become negative. This implies that the newly formed correlation matrix might be negative semi-definite, which violates the positive semi-definite property of correlation matrix. To avoid the aforementioned problem, a generalized deflation method is proposed in the work of Mackey[48], which obtains pseudo eigenvectors by maximizing the "additional" variance it captures.

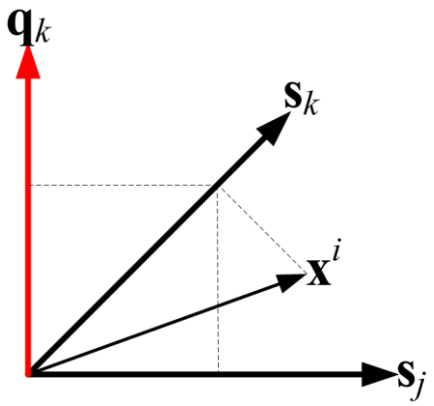

Figure 3-3: Variance removal example.

In Figure 3-3, $\mathbf{s}_j$ and $\mathbf{s}_k$ are non-orthogonal pseudo eigenvectors. The variance of the standardized data samples $\{\mathbf{x}^i\}_{i=1}^N$ in the direction of $\mathbf{s}_k$ is computed as $\sigma_k^2 = \frac{1}{N-1}\sum_{i=1}^N\langle\mathbf{x}^i,\mathbf{s}_k\rangle^2$. As $\mathbf{s}_k$ is not orthogonal to $\mathbf{s}_j$, part of σ_k is also captured by $\mathbf{s}_j$. The portion of σ_k on $\mathbf{s}_j$ is calculated as:

$$\sigma_k^2(\in \mathbf{s}_j) = \frac{1}{N-1}\sum_{i=1}^{N}\left\langle \mathbf{x}_i\mathbf{s}_k\mathbf{s}_k, \mathbf{s}_j \right\rangle^2. \tag{3.11}$$

The $\sigma_k(\in \mathbf{s}_j)$ is also removed in the standard Hotelling's deflation procedure, leading to the removal of excessive variance. The additional variance captured by $\mathbf{s}_k$, that is not on $\mathbf{s}_j$, lies in the direction orthogonal to $\mathbf{s}_j$, $\mathbf{q}_k = \frac{\mathbf{s}_k - \mathbf{s}_k^T\mathbf{s}_j\mathbf{s}_j}{||\mathbf{s}_k - \mathbf{s}_k^T\mathbf{s}_j\mathbf{s}_j||_2}$.

$$\sigma_k^2(\perp \mathbf{s}_j) = \frac{1}{N-1}\sum_{i=1}^{N}\left\langle \mathbf{x}_i, \mathbf{q}_k \right\rangle^2. \tag{3.12}$$

where $\sigma_k(\perp \mathbf{s}_j)$ is defined as the additional variance captured by $\mathbf{s}_k$ with respect to $\mathbf{s}_j$. In turn, the additional variance captured by a third pseudo eigenvector $\mathbf{s}_l$ with respect to $\mathbf{s}_j$ and $\mathbf{s}_k$ should lie in the direction that is orthogonal to both $\mathbf{s}_j$ and $\mathbf{s}_k$, $\mathbf{q}_l = \frac{\mathbf{s}_l - \mathbf{s}_l^T\mathbf{q}_k\mathbf{q}_k - \mathbf{s}_l^T\mathbf{s}_j\mathbf{s}_j}{||\mathbf{s}_l - \mathbf{s}_l^T\mathbf{q}_k\mathbf{q}_k - \mathbf{s}_l^T\mathbf{s}_j\mathbf{s}_j||_2}$. Let $\mathbf{s}_j = \mathbf{q}_j$, the process of finding the direction of additional variance for each pseudo eigenvectors is essentially a sequential Gram–-Schmidt process. The sequential Gram—Schmidt process is then adapted into the standard Hotelling's deflation method to obtain a set of pseudo eigenvectors that explain the most variance in data. At step *t* of the generalized deflation method, the additional variance captured by **s** is given as:

$$\begin{aligned}
\sigma_{\mathbf{q}}^2 &= \frac{1}{N-1}\sum_{i=1}^{N}\left\langle \mathbf{x}_i, \frac{\mathbf{s} - \mathbf{s}\sum_{l=1}^{t-1}\mathbf{q}_l\mathbf{q}_l^T}{\|\,\mathbf{s} - \mathbf{s}\sum_{l=1}^{t-1}\mathbf{q}_l\mathbf{q}_l^T\,\|_2} \right\rangle^2 \\
&= \frac{1}{N-1}\sum_{i=1}^{N}\left\langle \mathbf{x}_i, \frac{\mathbf{q}}{\|\,\mathbf{q}\,\|_2} \right\rangle^2 \\
&= \frac{\mathbf{q}^T\boldsymbol{\Sigma}^0\mathbf{q}}{\mathbf{q}^T\mathbf{q}}.
\end{aligned} \tag{3.13}$$

To obtain the pseudo-eigenvector, Eq. (3.13) is maximized under the following constraint.

$$\mathbf{s}_t = \underset{\mathbf{q}^T\mathbf{q}=1, \|\mathbf{s}\|_2=1, \|\mathbf{s}\|_0 \le k}{\arg\max}\ \mathbf{q}^T\boldsymbol{\Sigma}^0\mathbf{q} \tag{3.14}$$

In a sequential setting, rewriting $\mathbf{q} = \mathbf{s} - \mathbf{s}\sum_{l=1}^{t-1}\mathbf{q}_l\mathbf{q}_l^T = \mathbf{s}(\mathbf{I} - \sum_{l=1}^{t-1}\mathbf{q}_l\mathbf{q}_l^T) = \boldsymbol{s}\prod_{l=1}^{t-1}(\mathbf{I} - \mathbf{q}_l\mathbf{q}_l^T)$ yields the generalized deflation procedure for sparse PCA:

1) Set $\boldsymbol{\Sigma}_0^0 = \boldsymbol{\Sigma}^0, \boldsymbol{\delta}_0 = \mathbf{I}$;

2) $\mathbf{s}_t = \underset{\mathbf{s}^T\boldsymbol{\delta}_{t-1}\mathbf{s}=\mathbf{1}, \|\mathbf{s}\|_0 \le k}{\operatorname{argmax}}\ \mathbf{s}^T\boldsymbol{\Sigma}_{t-1}^0\mathbf{s}$;

3) $\mathbf{q}_t = \boldsymbol{\delta}_{t-1}\mathbf{s}_t$;

4) $\boldsymbol{\Sigma}_t^0 = (\mathbf{I} - \mathbf{q}_t\mathbf{q}_t^T)\boldsymbol{\Sigma}_{t-1}^0(\mathbf{I} - \mathbf{q}_t\mathbf{q}_t^T)$;

5) $\boldsymbol{\delta}_t = \boldsymbol{\delta}_{t-1}(\mathbf{I} - \mathbf{q}_t\mathbf{q}_t^T)$;

6) $\mathbf{s}_t = \frac{\mathbf{s}_t}{||\mathbf{s}_t||_2}$.

The above procedure is iterated until a predefined number of pseudo eigenvectors are obtained.

3.3.2 Selection of the Sparsity parameter k

The pseudo-eigenvector with an appropriate level of sparsity can reveal meaningful features from the process data. In this study, a parameter selection approach based on evaluating the total amount of variance captured by pseudo-vectors with different levels of sparsity is proposed. In detail, the sparsity parameter *k* is increased in increments of 1 from 1 to be equal to the total number of process variables *d* (from completely sparse to non-sparse), $k = \{1,2,3,\dots,d\}$. For each of the *k* values, a set of *d* pseudo-eigenvectors are obtained. The variance associated with the first sparse pseudo-eigenvector, $\mathbf{s}_1 = \mathbf{q}_1$, is determined.

$$\sigma_{\mathbf{q}_1}^2 = \frac{1}{N-1}\sum_{i=1}^{N}\langle \mathbf{x}_i, \mathbf{s}_1 \rangle^2. \tag{3.15}$$

Subsequently, the total amount of variance captured by the first two pseudo-eigenvectors is the sum of $\sigma_{\mathbf{q}_1}^2$ and the additional variance captured by the second pseudo-eigenvector. By induction, the total amount of variance captured by all the pseudo-eigenvectors with sparsity *k* is computed as following.

$$\Xi_{||\mathbf{s}||_0 \le k}^2 = \sigma_{\mathbf{q}_1}^2 + \sum_{2}^{d}\sigma_{\mathbf{q}_i}^2. \tag{3.16}$$

where $\sigma_{\mathbf{q}_i}^2$is determined using Eq. (3.13). Finally, a parameter selection plot with y-axis being values of $\sigma_{||\mathbf{s}||_0 \le k}^2$and x-axis being the values of *k* is generated. The appropriate value of *k* is determined to be the one after which any further increase does not yield significant increase in the total amount of variance captured by the pseudo-eigenvectors.

The magnitude of each entry of the eigenvector gives an indication the importance of the corresponding process variable; the process variable with an entry of large magnitude contributes significantly to the total variance explained in the direction of the eigenvector. In practice, it may only require very few of these important process variables to explain the same amount of variance as explained by all process variables. The proposed parameter selection approach incrementally relaxes the sparsity constraint (by increasing value of *k*) and reveals the important process variables until the total amount of variance

explained does not increase further. In this respect, the selected *k* value ensures that sparse eigenvectors are obtained with minimum information loss.

3.4 Online fault diagnosis

The pseudo eigenvectors generated in Section 3.3 span the normal subspace for online fault diagnosis. This subspace is only valid if and only if all the pseudo eigenvectors are independent of each other. Such a condition is proven in the following Theorem.

Theorem 1. *The pseudo eigenvectors extracted using the generalized deflation method are always independent of each other.*

Proof. Suppose *r* number of pseudo eigenvectors, $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_r\}$ have been extracted using the generalized deflation method.

$$\begin{aligned}
&\mathbf{q}_1 = \frac{\mathbf{s}_1}{\| \mathbf{s}_1 \|_2};\\
&\hat{\mathbf{q}}_2 = \mathbf{s}_2 - \mathbf{q}_1^T \mathbf{s}_2 \mathbf{q}_1;\\
&\mathbf{q}_2 = \frac{\hat{\mathbf{q}}_2}{\| \hat{\mathbf{q}}_2 \|_2};\\
&\mathbf{s}_2 = \mathbf{q}_1^T \mathbf{s}_2 \mathbf{q}_1 + \| \hat{\mathbf{q}}_2 \|_2 \, \mathbf{q}_2;\\
&\bullet\\
&\bullet\\
&\bullet\\
&\hat{\mathbf{q}}_r = \mathbf{s}_r - \mathbf{q}_1^T \mathbf{s}_r \mathbf{q}_1 - \mathbf{q}_2^T \mathbf{s}_r \mathbf{q}_2 - \dots - \mathbf{q}_{r-1}^T \mathbf{s}_r \mathbf{q}_{r-1};\\
&\mathbf{q}_r = \frac{\hat{\mathbf{q}}_r}{\| \hat{\mathbf{q}}_r \|_2};\\
&\mathbf{s}_r = \hat{\mathbf{q}}_r + \mathbf{q}_1^T \mathbf{s}_r \mathbf{q}_1 + \mathbf{q}_2^T \mathbf{s}_r \mathbf{q}_2 + \dots + \mathbf{q}_{r-1}^T \mathbf{s}_r \mathbf{q}_{r-1};
\end{aligned} \tag{3.17}$$

By induction, for $t \in \{1,2, \dots, r\}$, $\mathbf{s}_t \in \text{span}\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_t\}$ and $\mathbf{q}_t \in \text{span}\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_t\}$. The last equality in Eq. (3.17) can be reorganized as:

$$\begin{aligned}
&\mathbf{q}_1 = \frac{\mathbf{s}_r}{\mathbf{q}_1^T \mathbf{s}_r} - \frac{\mathbf{q}_2^T \mathbf{s}_r}{\mathbf{q}_1^T \mathbf{s}_r} \mathbf{q}_2 - \dots - \frac{\mathbf{q}_{r-1}^T \mathbf{s}_r}{\mathbf{q}_1^T \mathbf{s}_r} \mathbf{q}_{r-1} - \frac{\| \hat{\mathbf{q}}_r \|_2}{\mathbf{q}_1^T \mathbf{s}_r} \mathbf{q}_r;\\
&\mathbf{q}_2 = \frac{\mathbf{s}_r}{\mathbf{q}_2^T \mathbf{s}_r} - \frac{\mathbf{q}_1^T \mathbf{s}_r}{\mathbf{q}_2^T \mathbf{s}_r} \mathbf{q}_1 - \dots - \frac{\mathbf{q}_{r-1}^T \mathbf{s}_r}{\mathbf{q}_2^T \mathbf{s}_r} \mathbf{q}_{r-1} - \frac{\| \hat{\mathbf{q}}_r \|_2}{\mathbf{q}_2^T \mathbf{s}_r} \mathbf{q}_r;\\
&\bullet\\
&\bullet\\
&\bullet\\
&\mathbf{q}_r = \frac{\mathbf{s}_r}{\| \hat{\mathbf{q}}_r \|_2} - \frac{\mathbf{q}_1^T \mathbf{s}_r}{\| \hat{\mathbf{q}}_r \|_2} \mathbf{q}_1 - \dots - \frac{\mathbf{q}_{r-1}^T \mathbf{s}_r}{\| \hat{\mathbf{q}}_r \|_2} \mathbf{q}_{r-1}.
\end{aligned} \tag{3.18}$$

Since $\mathbf{q}_t \in \text{span}\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_t\}$, each $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_r\}$ is a linear combination of $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_r\}$. Therefore, $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_r\}$ form the basis for $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_r\}$ which is an orthonormal basis of a subspace $\mathbb{R}^r$. This implies that $\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_r\}$ have to span $\mathbb{R}^r$ as well—$\{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_r\}$ have to be independent of each other. The next crucial step is to determine the appropriate number of pseudo-eigenvectors for online process monitoring. A modified scree plot is proposed to achieve this. As compared to the traditional scree plot, the fraction/percentage of the variance explained by each pseudo eigenvector is computed as the following ratio.

$$\text{Fraction of variance explained} = \frac{\sigma_{\mathbf{q}_i}^2}{\Xi_{\|\mathbf{s}\|_0 \le k}^2} \tag{3.19}$$

According to Proposition 1, the amount of variance associated with each true eigenvector is equal to its true eigenvalue, in an analogous way, the additional variance associated with each pseudo-eigenvector is referred to as its pseudo-eigenvalue. The pseudo-eigenvectors can be ordered based on the magnitude of their pseudo-eigenvalues. The application of this modified scree plot in determining the appropriate number of pseudo-eigenvectors is demonstrated in the second case study. During real-time monitoring of a process, an on-line data sample is first transformed and then projected into the subspace spanned by $\mathbf{S}_r = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_r\}$ to form a new vector.

$$\mathbf{y}^i = \mathbf{Q}_r^T \left(\mathbf{x}^i\right)^T. \tag{3.20}$$

where $\mathbf{y}^i \in \mathbb{R}^r, r \le d$. Subsequently, based on Remark 1, the T^2 and SPE statistics could be calculated in the similar way of [15].

$$T_i^2 = \left(\mathbf{y}^i\right)^T \mathbf{\Lambda}_r^{-1} \mathbf{y}^i. \tag{3.21}$$

where Λ_r is a diagonal matrix whose diagonal elements are the first r eigenvalues associated with $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_r\}$. Although the pseudo eigenvectors are independent to each other, they are not orthogonal to each other. In the case, the orthonormal vectors $\{\mathbf{q}_1, \mathbf{q}_2, \dots, \mathbf{q}_r\}$ are used to compute the SPE statistic.

$$f(\mathbf{x}^i) = \mathbf{Q}_r \mathbf{Q}_r^T \left[f(\mathbf{x}^i) \right]^T. \tag{3.22}$$

$$SPE_i = \left[\left(\mathbf{x}^i\right)^T - \mathbf{x}^i \right]^T \left[\left(\mathbf{x}^i\right)^T - \mathbf{x}^i \right]. \tag{3.23}$$

The control limits for both statistics are estimated using Kernel Density Estimation.[62] For fault diagnosis, T^2 statistic can be decomposed to identify the individual contribution of each process variable.

$$\begin{aligned} T_i^2 &= \left(\mathbf{y}^i\right)^T \boldsymbol{\Lambda}_r^{-1}\mathbf{y}^i; \\ &= \left(\mathbf{y}^i\right)^T \boldsymbol{\Lambda}_r^{-1}\mathbf{Q}_r^T\left(\mathbf{x}^i\right)^T; \\ &= \sum_{j=1}^{d}\left(\mathbf{y}^i\right)^T \boldsymbol{\Lambda}_r^{-1}\mathbf{w}_j x_j^i; \\ &= \sum_{j=1}^{d} cont_j \end{aligned} \tag{3.24}$$

Where $\mathbf{w}_j$ is the j^{th} column of $\mathbf{Q}_r^T$. Similarly, SPE statistic can be decomposed as:

$$cont_j(SPE_i) = \left(x_j^i - x_j^i\right)^2. \tag{3.25}$$

3.5 Case Studies

Two industrial case studies are used to evaluate the performance of the proposed sparse PCA. The first case study is a simulation of the Continuous Stirred Heating Tank (CSHT)[63] and the second case study is the benchmark Tennessee Eastman chemical process (TE process).[32] For both case studies, the process monitoring performances of the proposed technique, PCA, KPCA, and KICA are compared under a contamination free setting. The reason that such a performance comparison is not made under data contamination is because although contaminated data may not have the exact same pattern as compared to the original data, its pattern could still differ significantly from that of the faulty data, leading to high fault detection rate. With the increase of data contamination rate, the contaminated data may possess less features of the normal process data and more features from the contamination, and become more distinctive to the faulty process; the fault detection rate could be further improved. Therefore, fault detection rate is not considered as an indicative measure of robustness of the proposed sparse PCA.

To obtain a comprehensive assessment of the robustness of the techniques presented in this work, three error terms are used. The first one is the mean squared errors between the correlation coefficients (for every pair of process variables) estimated from the contamination free data and from the contaminated data under different contamination rate. It assesses the accuracy of the Pearson's, Spearman' rank, and Kendall tau's rank correlation measures in correlation coefficient estimation.

$$MSE = \frac{\sum_{i=1}^{d}\sum_{j=1}^{d}\left(\rho_{ij}^0 - \hat{\rho}_{ij}^{\varepsilon}\right)^2}{d \times d - d} \tag{3.26}$$

where ρ_{ij}^0 and $\hat{\rho}_{ij}^{\varepsilon}$ are the correlation coefficients (Pearson, Spearman, and Kendall tau) between process variables X_i and X_j, estimated from the contamination free data and the contaminated data with contamination rate ε, respectively.

The second error terms is proposed by Boudt, et al.[46] to analyze the robustness of the correlation measures to data contamination. It determines the minimum contamination rate at which the sign of the estimated correlation coefficient is inverted or becomes non-informative. In this study, all the correlation coefficients are considered at the same time and the percentage of inverted correlation coefficients under different data contamination rate is used to assess the robustness of the correlation measures.

$$\delta_{ij}^{\varepsilon} = \begin{cases} 0 & \rho_{ij}^{0}\hat{\rho}_{ij}^{\varepsilon} > 0 \\ 1 & \rho_{ij}^{0}\hat{\rho}_{ij}^{\varepsilon} \le 0 \end{cases}; \tag{3.27}$$

$$\text{Percentage of inverted correlation coefficients} = \frac{\sum_{i=1}^{d}\sum_{j=1}^{d}\delta_{ij}^{\varepsilon}}{d \times d} \times 100\% \tag{3.28}$$

The last error term evaluates the robustness of the studied techniques in feature extraction. Specifically, the first eigenvector or pseudo-eigenvector extracted from a correlation matrix computed using contaminated data under a range of contamination rates are compared with their counterparts in contamination free condition. The difference between the eigenvectors is measured by the angle between them.

$$\theta_{\varepsilon} = \cos^{-1}\left(\frac{\mathbf{v}_1\hat{\mathbf{v}}_1^{\varepsilon}}{\|\mathbf{v}_1\|\|\hat{\mathbf{v}}_1^{\varepsilon}\|}\right) \tag{3.29}$$

where θ_{ε} is the angle between the eigenvector obtained from data under contamination rate ε, $\hat{\mathbf{v}}_1^{\varepsilon}$, and the original eigenvector, $\mathbf{v}_1$, under no data contamination. This error term measures the ability of the PCA techniques to extract accurate latent features from contaminated process data.

A point mass contamination procedure shown in Eq. (3.30) is used to contaminate the normal process data for robustness evaluation.

$$F_j = (1-\varepsilon)F_j^0 + \varepsilon\delta_1 \tag{3.30}$$

where F_j^0 is the distribution of the uncontaminated data of the j^{th} process variable, $\delta_1$is a point mass distribution at 1, and F_j is the contaminated distribution. The parameter ε is used to control the percentage of data contamination.

3.5.1 Continuous stirred tank heater

The Continuous stirred heating tank (CSHT) simulation was first developed by Thornhill, et.al [63]. The CSHT consists of a heating tank in which hot and cold water are uniformly mixed. The tank is heated with steam running through a heating coil. Three variables of the process are closed-loop controlled: tank level, cold water supply flow rate, and the outlet flow temperature. In addition, these three variables are also constantly

subjected to random disturbances. The schematic diagram of the CSHT is shown in Figure 3-4.

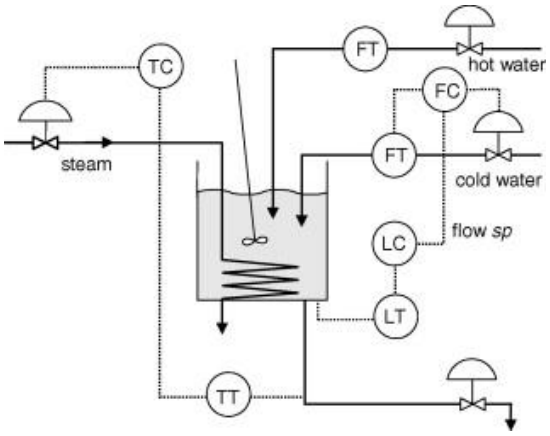

Figure 3-4: Schematic diagram of the continuous stirred heating tank.

A detailed description of the CSHT model can be found at http://personal-pages.ps.ic.ac.uk/~nina/CSTHSimulation/index.htm. In this case study, the three controlled process variables are monitored in real time. Since there is only a small number of process variables, the proposed sparse PCA is not applied to this case study. Instead, a regular PCA technique is utilized to extract non-sparse eigenvectors from correlation matrices formed using Pearson's, Spearman's rank and Kendall tau's correlation measures, respectively. Due to the similar reason, KPCA and KICA are not tested on this simple case study either. The main goal of this simple case study is then to establish the basic viability of the Spearman's rank and Kendall tau's correlation measures in nonlinear process monitoring and robust extraction of non-sparse features. Once this viability is confirmed, a more complicated TE process involving 33 process variables is used to further evaluate the effectiveness of the proposed sparse PCA technique. The PCA techniques involving computing the Spearman's rank and Kendall tau's correlation matrices are referred to as robust nonlinear PCA or RNPCA in the context of present study, while its sparse version is called the robust nonlinear sparse PCA or RNSPCA.

One hundred normal data samples with 0% contamination are collected to construct the standard scale-invariant PCA and the RNPCA models for process monitoring. A standard scree plot shown in Figure 3-5 is used to determine the appropriate number of eigenvectors.

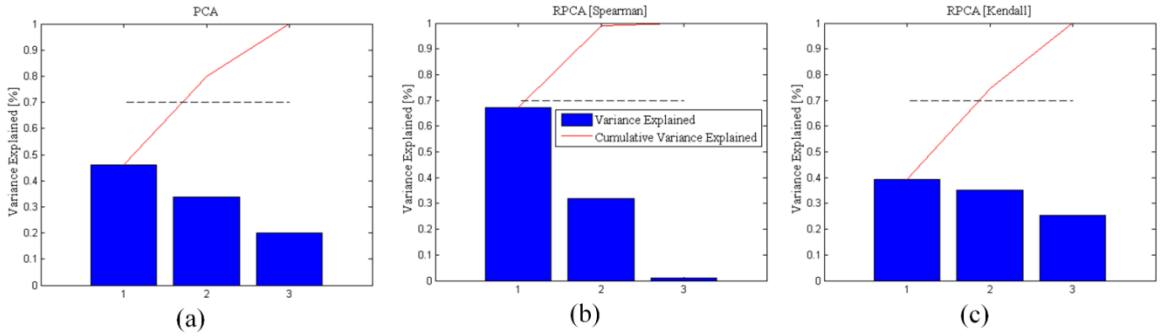

Figure 3-5: scree plot for determination of number of non-sparse eigenvectors.

As can be seen in Figure 3-5, the first two eigenvectors for all three techniques are able to explain more than 70% of data variance. Therefore, 2 eigenvectors are retained for these technique to construct the feature space. The cut-off point of explaining 70% variance is also used for the TE process case study. The CSHT system is monitored for 200 sample intervals. There are two fault conditions generated at sample interval 100. The first fault condition is introduced as an increased random variation (Gaussian noise with zero mean and 0.05 variance) to the temperature measurement. The second fault condition is the beta noise with parameters 4 and 1 imposed on the flow measurement. Both fault conditions are fed to the PID controller as feedback signal to upset the process operation. The fault detection rate (FDR) and false alarm rate (FAR) for both PCA and the RNPCA techniques are presented in Table 3-2.

Table 3-2: Process monitoring results of the CSHT case study

	FDR				FAR			
	Fault 1		Fault 2		Fault 1		Fault 2	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
PCA	64	44	61	60	2	3	3	2.5
RNPCA [Spearman]	77	46	70	70	2	2	2	2
RNPCA [Kendall]	77	41	70	71	2	2	2	3

In addition, the process monitoring charts for both fault conditions are shown in Figure 3-6 and Figure 3-7.

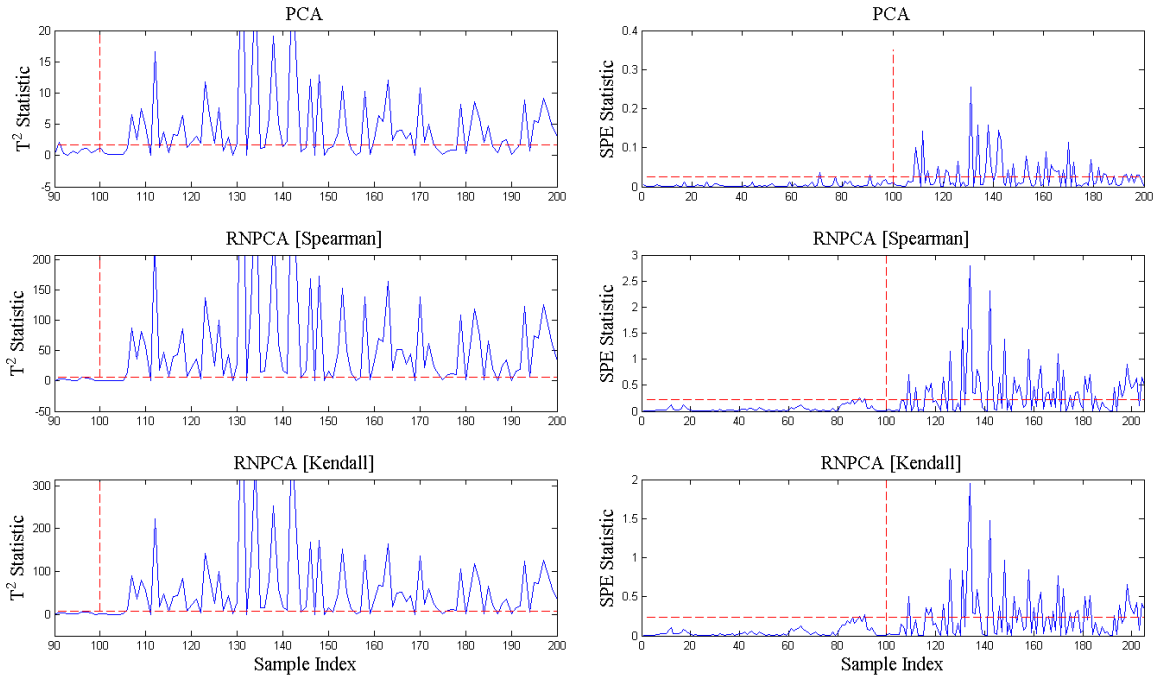

Figure 3-6: Process monitoring charts of the Fault 1 of the CSHT process.

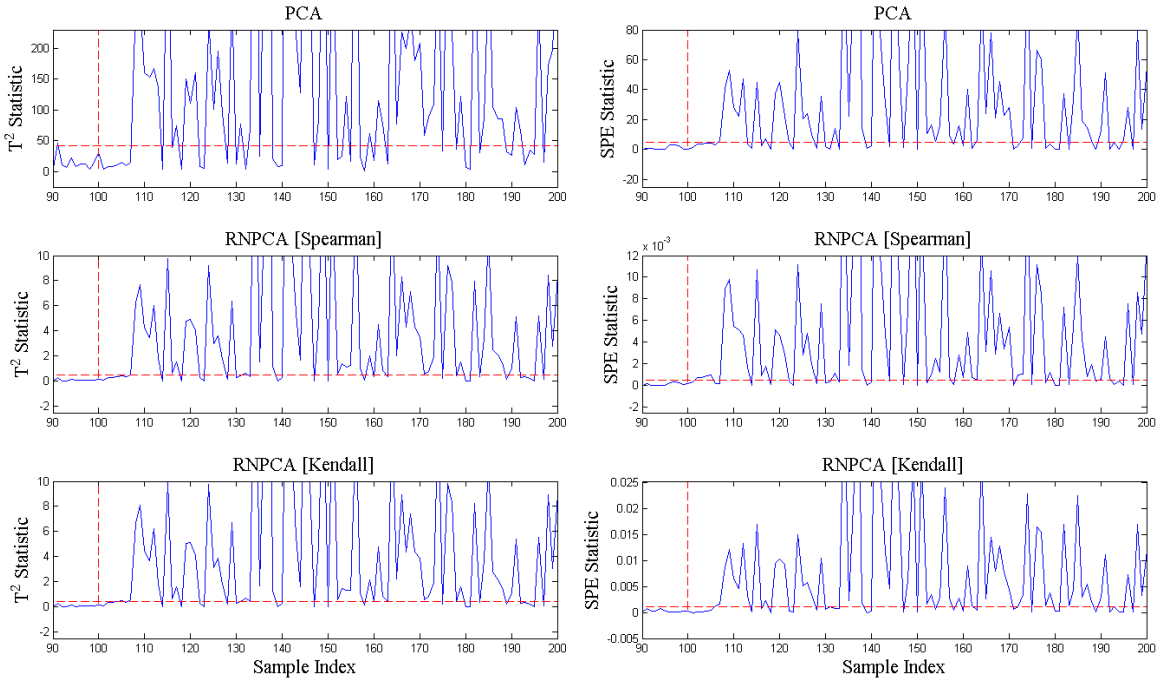

Figure 3-7: Process monitoring charts of the Fault 2 of the CSHT process.

T^2 statistic measures the systematic variation of the system. It is computed as a Mahalanobis distance in the subspace spanned by the eigenvectors. Conversely, SPE statistics quantifies the amount of residual variation that is not explained in the PCA model (explained in the Null space of the feature space). These two statistics complement each other for process monitoring. For example, a large increase in T^2 statistics but not in SPE statistics indicates a fault condition that causes large systematic variation but

preserves the correlation structure. In the situation where both statistics increase significantly indicates a fault condition violating the correlation structure.

For this case study, T^2 statistics of the RNPCA techniques perform the best, with more than 70% detection rate for both fault conditions. With regard to standard PCA, the eigenvectors are obtained by decomposing the Pearson's correlation matrix which only stores linear relationships between the process variables. In this respect, the PCA eigenvectors retain only the linear correlation structure. In Fault 1, if the system behaves linearly, the increased Gaussian variation introduced should be captured as systematic variation; thus leading to large increase of the T^2 statistic. However, the CSHT model is highly nonlinear by design which further disrupts the introduced Gaussian disturbance. The resulting disturbance is neither linear (violating linear structure) nor Gaussian and cannot be completely distinguished in the feature space. In principal, the unexplained variation should all fall into the residual space which is measured by the SPE statistic. The SPE statistic is in essence the reconstruction error from the normal subspace back to the original measurement space. SPE statistics of both the standard PCA and the RNPCA capture a large amount of non-Gaussian residual information.

In Fault 2, with the introduction of beta noise (non-Gaussian), fault detection rate of T^2 statistics of the proposed method declines slightly to 70% but still achieving better result compared to that of the standard PCA, indicating the retention of more nonlinear information. On the other hand, the performances of SPE statistic for all the techniques are at almost the same level with the T^2 statistic. This is due to the fact none of these techniques are able to explain non-Gaussian variations in the feature space. The unexplained variations are captured in the residual space. This limitation of the RNPCA can be greatly relaxed when monitoring large number process variables simultaneously, as a virtue of the central limit theorem (shown in the TE process case study). Additionally, a comparison between the RNPCA adopting Kendall tau's and Spearman's correlation matrices demonstrates that these two correlation measures are equally potent in modelling nonlinear dependence structure.

The fault diagnosis results for both fault conditions are shown in Figure 3-8 and Figure 3-9, respectively.

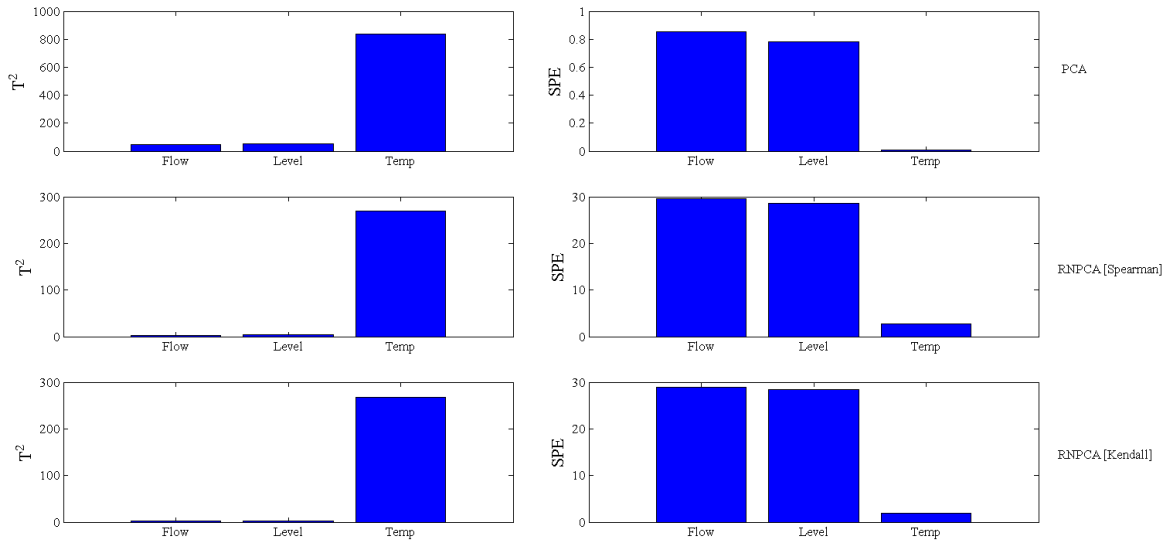

Figure 3-8: Variable contribution charts for Fault 1.

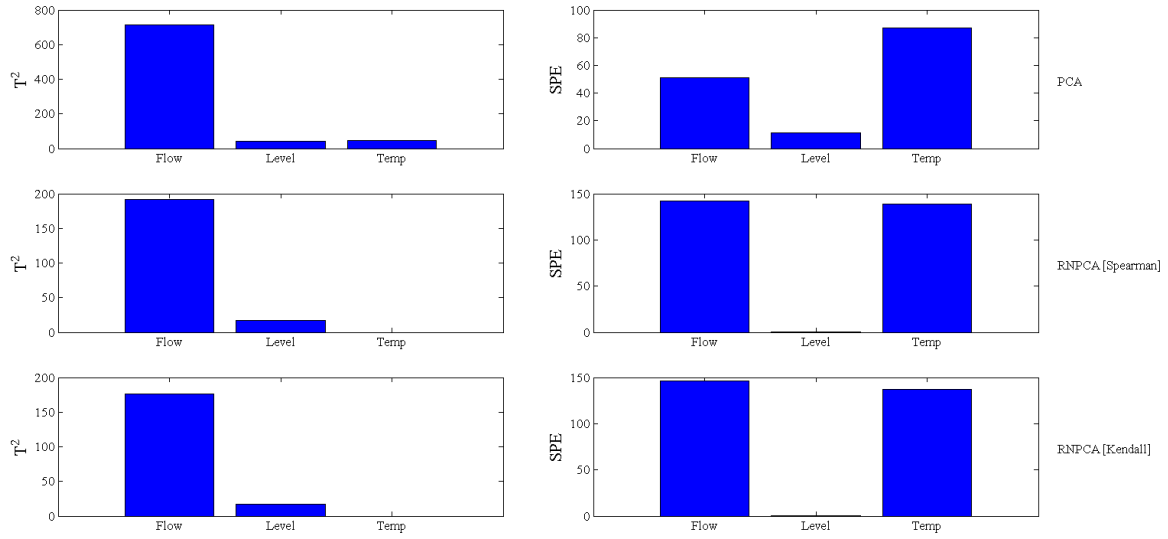

Figure 3-9: Variable contribution charts for Fault 2.

The contribution plots are generated by summing the contribution of each process variable to first 10 T^2 and SPE statistics after the introduction of fault. For Fault 1, T^2 contribution plots for both methods are able to identify the correct root-cause variable (temperature). Conversely, the SPE statistics are unable to locate the root-cause variable. For Fault 2, the standard PCA offers a similar performance as in Fault 1—T^2 statistic made the correct diagnosis but SPE statistic did not. Both T^2 and SPE statistics of the proposed methods accurately locates the true root-cause (flow), owing to their abilities to retain nonlinear correlation structure in the eigenvectors.

To determine the robustness of the PCA techniques considered in this case study, normal process data with different contamination levels are used for testing. The probability density functions (PDF) of the process variables with 25% point-mass contamination are shown in Figure 3-10.

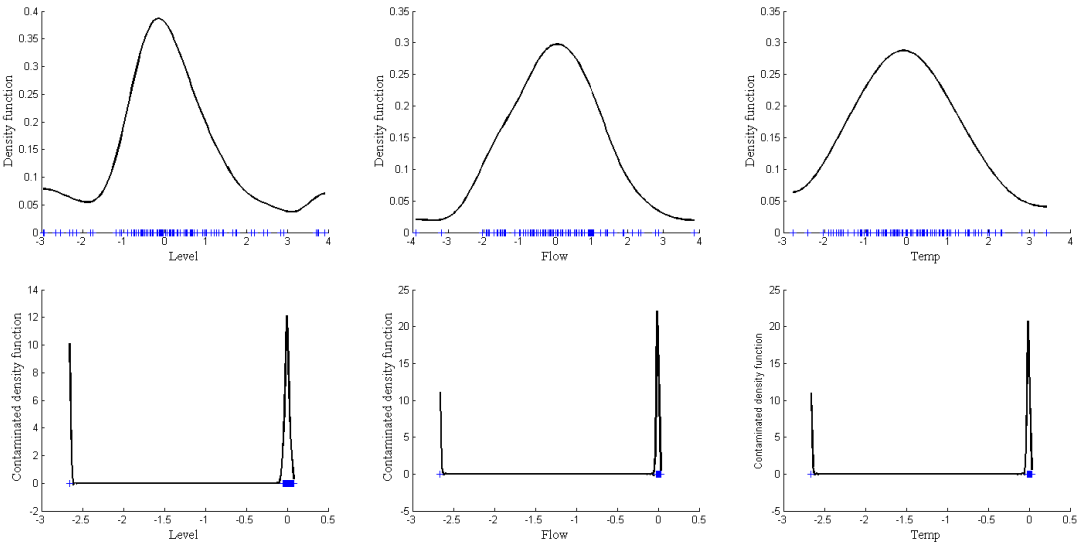

Figure 3-10: Probability density functions of the process variables of CSHT under 25% point-mass data contamination.

These PDFs are estimated using kernel density estimation.[62] The large spike in the tail region of the contaminated PDFs is due to the fact that 25% of the data points are sampled from the point-mass (Dirac) distribution at 1. The results for the Angle and MSE values of all the techniques from 5% to 60 % contamination rate are shown in Figure 3-11.

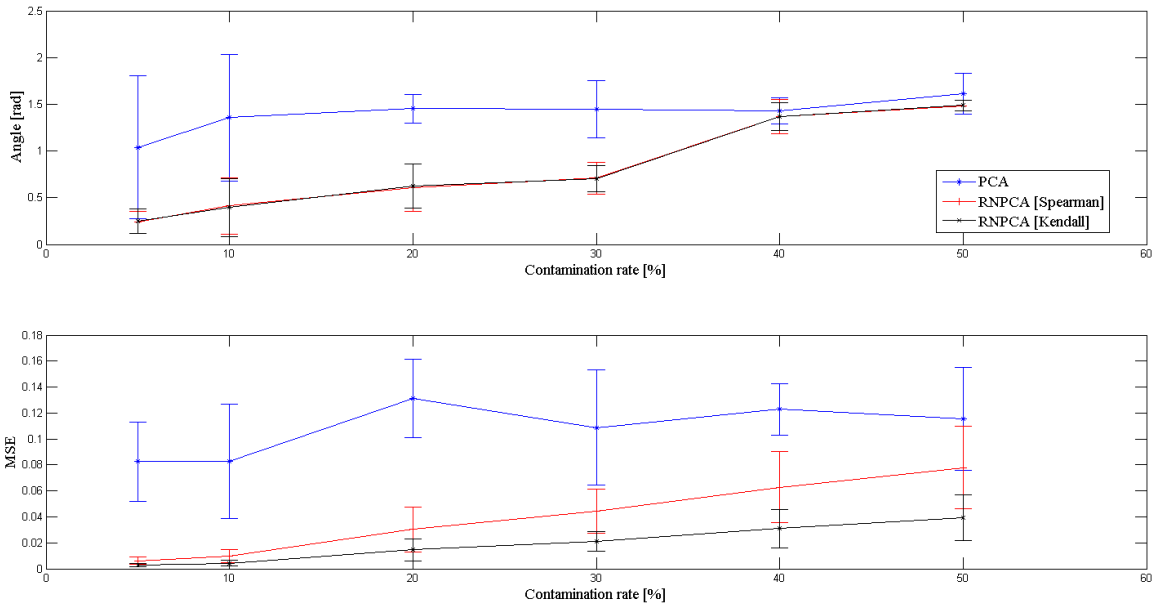

Figure 3-11: Angle and MSE values for robustness testing of CSHT case study.

The error bar at each contamination rate is obtained by repeating the simulation 20 times. It is clearly shown that accuracy of the Pearson's correlation is compromised at a very low contamination rate (5%). The magnitude of the MSE and Angle stay almost the same throughout the range of the contamination rates. Conversely, the Spearman's and Kendall tau's correlation measures are demonstrated to be more robust under

contamination. However, with the increase in contamination rate, their performances deteriorate slowly and eventually reach at the same level with Pearson's correlation measure at 50% contamination rate for the Angle error term. For the MSE error term, the Kendall tau's rank correlation measure is shown to be more robust than the Spearman's rank correlation measure. This results comply with the conclusions drawn in the work of Croux, Dehon[51]. The percentage of inverted correlation coefficients under different data contamination rates is not determined in this simple case study as there are only 100 normal data samples for testing, which is not adequately large to achieve a consistent result. It is therefore confirmed in this simple case study that the proposed method provides better performance in nonlinear process monitoring. It is also revealed that the use of Spearman's and Kendall tau's rank correlation measures makes the RNPCA much more robust.

3.5.2 Tennessee Eastman process

In this case study, the proposed techniques is tested on the Tennessee Eastman process. Its performance is compared with the performances of PCA, KPCA and KICA under data contamination free condition. Similar to the first case study, normal process data having different levels of data contamination are used to test and verify the robustness of the proposed RNPCA techniques. This testing is also extended to the RNSPCA to assess its sparse feature discovery performance under data contamination.

The Tennessee Eastman process comprises of five major operating units: an exothermic two-phase reactor, a product condenser, a vapor-liquid flash separator, a recycle compressor, and a reboiled product stripper. The process flow diagram of the chemical plant is shown in Figure 9-1 in the Appendices. A detailed explanation of this benchmark process can be found in the work of Downs, Vogel[32]. In total there are 44 measured process variables. Thirty-three of these process variables, including 22 continuously monitored process variables and 11 manipulated process variables, are monitored for online fault diagnosis. These variables are listed in Table 9-1.

In the first part of this case study, 960 normal data samples with 0% contamination are used to construct subspace models for PCA, KPCA, KICA, RNPCA, and RNSPCA for process monitoring. In the second part of this study, the percentage of the contamination will increase from 5% to 80% to verify whether Kendall tau's and Spearman's rank correlation measures are more robust than Pearson's correlation measure under a high dimensional setting.

For KPCA and KICA, the parameters for the Radial Basis kernel are determined using the methods described in.[18,20] The number of eigenvectors for the PCA is determined to be 14 through cross-validation using the contamination free training data. The scree plot for this determination is shown in Figure 3-12 in which the cut-off percentage of variance explained is set to 70%. In addition, the scree plot for the robust PCA (Spearman and Kendall) and the modified scree plot for the sparse PCA (Kendall, $k = 4$) are also shown in the same figure.

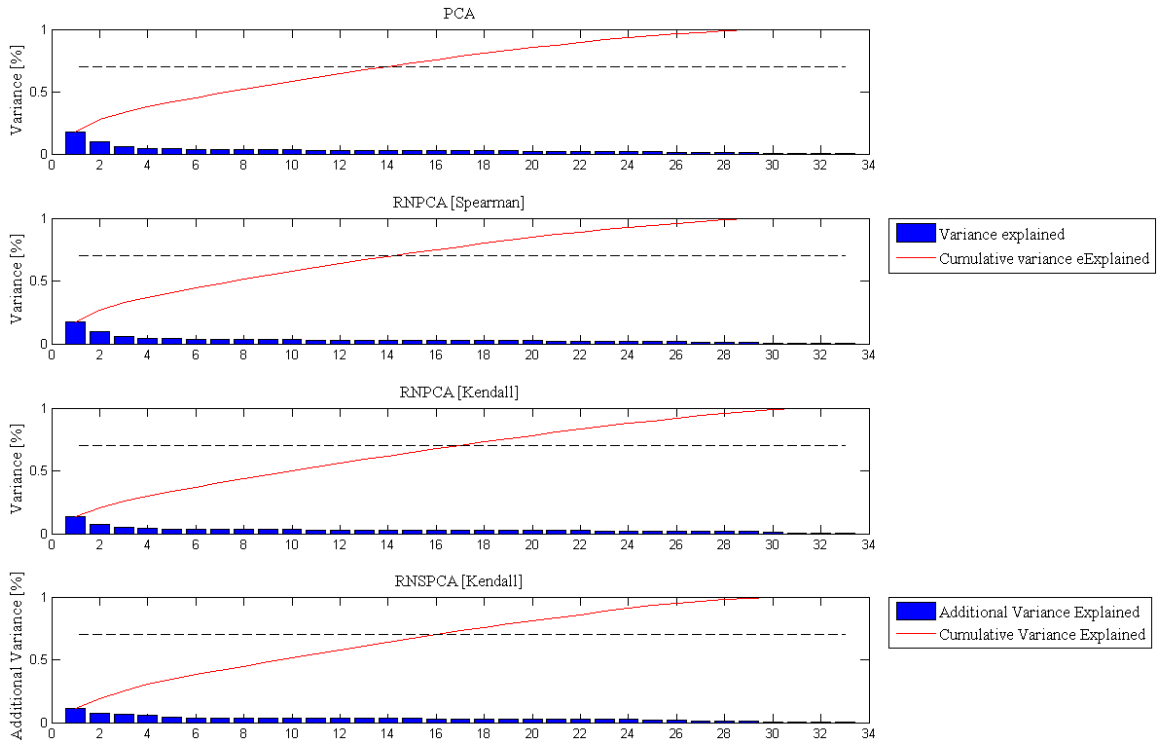

Figure 3-12: Scree plots for PCA, RNPCA [Spearman], RNPCA [Kendall], and RNSPCA [Kendall].

It is observed that the RNSPCA requires 16 PCs to explain the same percentage of variance. This is due to the fact that the sparse eigenvectors are not the true eigenvectors of the correlation matrix. To establish a consistent basis for comparison, the number of latent variables for all the techniques are set to 14.

For the RNSPCA model, the appropriate number of sparsity parameter k is determined for both correlation measures using the method outlined in Section 3-12 under 0% data contamination. The determined sparsity number k is also used for data contamination case. A modified cross-validation plot based on additional variances captured for both correlation measures is presented in Figure 3-13.

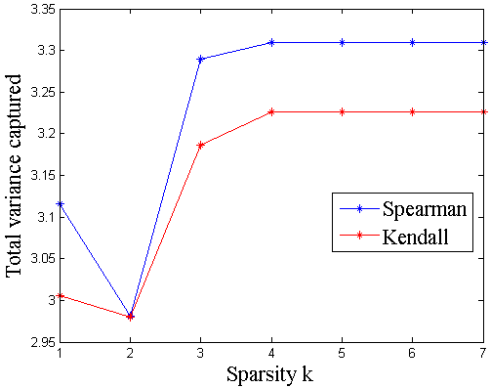

Figure 3-13: Cross-validation plot for sparsity k selection.

It is observed that both correlation measures are able to capture almost the same amount of variances. As k increases, the amount of captured variance decreases from k =

1 to $k = 2$ and then increases from $k = 2$ to $k = 4$. This trend eventually reaches a plateau after $k = 4$. In this respect, the number of k is selected to be 4. This cross-validation procedure is carried out using Eq. (3.5). To obtain the same number of sparsity k, the L_1 parameter β in Eq. (3.6) is also tuned from 1 to d in increments of 1. The appropriate number of β is determined to be 2. The first pseudo-eigenvectors obtained from both optimization methods are listed in Table 3-3.

Table 3-3: First pseud-eigenvectors determined using SDP relaxation and L_1 relaxation [Kendall]

Variable No.	First true eigenvector (PCA)	First Pseudo-eigenvector fmincon (Kendall)	First Pseudo-eigenvector CVX (Kendall)
X1	-0.1035	-0.0052	0.0003
X2	-0.0057	0.0032	0.001
X3	0.1398	0.0013	0.0038
X4	0.0343	0.0004	0.0008
X5	-0.0012	-0.0005	0.0001
X6	-0.0325	-0.0013	0.0034
X7	0.3684	**0.5533**	**0.7494**
X8	0.0751	-0.0046	0.0032
X9	0.0041	-0.0051	0.0025
X10	-0.1656	-0.0009	0.0035
X11	-0.2636	-0.1138	0.0095
X12	0.0127	-0.0033	0.0011
X13	0.3659	**0.5511**	**0.6543**
X14	-0.0273	-0.0016	0.0004
X15	-0.0117	0.0014	0.0013
X16	0.3632	**0.5280**	**0.0996**
X17	-0.0090	-0.0011	0.0003
X18	-0.3286	-0.1382	0.0034
X19	-0.2760	-0.0224	0.0021
X20	0.1616	0.0993	0.0019
X21	-0.0218	0.0005	0.0098
X22	-0.0732	0.0029	0.0022
X23	-0.0092	-0.0011	0.0009
X24	-0.1089	-0.0063	0.0004
X25	0.2554	0.0020	0.0013
X26	-0.0250	-0.0018	0.0023
X27	-0.1482	-0.0037	0.0009
X28	-0.0671	-0.0054	0.0002
X29	-0.0008	0.0046	0.0042
X30	-0.1263	0.0008	0.0014
X31	0.2961	**0.2347**	0.0003
X32	-0.0464	-0.0042	0.0004
X33	-0.19033	-0.0347	0.0018
First eigenvector	Variance = 9.1621 Variance [%] = 17.58%	Variance = 5.6532 Variance [%] = 11.18%	Variance = 3.3093 Variance [%] = 100%
All eigenvectors	Total variance = 52.1166	Total variance = 50.5653	Total variance = 3.2262

It is demonstrated both optimization methods are able to yield meaningful sparse eigenvectors. The loadings of these two pseudo-eigenvectors are almost zero except for those of the process variable X7 (reactor pressure), X13 (Separator pressure), X16 (Stripper pressure) and X31 (Stripper steam valve). Each of these process variables is

associated with a major operating unit of the process. Both the stripper and the separator have return streams back to the reactor. The pressure of stripper and the separator has the most effect among all the process variables on the reactor's operating condition. In this regard, it is not difficult to see that pressure in each of these major operating units play pivotal role of integrating the operations and ensuring all the process units function as a whole. The loadings of the sparse eigenvector provide meaningful interpretation of the importance of each process variable in this case study. It is also observed that the total amount of variance captured by all the eigenvectors between the standard PCA and the RNSPCA based on the fmincon package are fairly close to each other. However, the first pseudo-eigenvector obtained using the CVX method captures all the variance in the feature space; the estimation is extremely biased. The total amount of variance captured for the CVX version is also significantly lower as compared to the other two techniques. Due to the above reasons, the RNSPCA relying on the fmincon package is used for the ensuing simulations. For the second part of this case study where contaminated data is used for training, the number of eigenvectors is also set to 14 for the sake of consistence in comparison. All of these techniques are implemented on the Matlab 2010b platform on a personal computer with Core i7-3740QM CPU at 2.70GHz and 16GB of RAM. The elapsed CPU time for training the listed techniques are presented in Table 3-4. To establish a consistent basis for comparison, the maximum number of iterations for extracting a single eigenvector for both the KICA and the RNSPCA based on fmincon package is set to 100. On the other hand, the PCA and KPCA techniques rely on eigenvalue decomposition which do not require an iterative optimization procedure. For the CVX package, the required number of iterations is automatically determined.

Table 3-4: Comparison of the computational time for training the considered techniques.

	CPU time [s] ± standard error	Maximum number of iterations
PCA	0.4524±0.0321	-
KPCA	4.8984±0.9678	-
KICA	223.3466±1.3845	100
RNSPCA [Kendall] [fmincon]	22.4017±0.4047	100
RNSPCA [Spearman] [fmincon]	17.8153±0.2103	100
RNSPCA [Kendall] [CVX]	181.6725±1.2724	Automatically determined in CVX package
RNSPCA [Spearman] [CVX]	196.4802±0.8319	Automatically determined in CVX package

It is observed KICA is the least efficient technique, requiring more than 200 seconds of CPU time for training. The proposed RNSPCA based on the CVX package is much slower than the RNSPCA relying on the fmincon package. This is probably due to use of numerical tolerance level as a stopping criteria in the CVX package. By default, there are three numerical tolerance levels that can be chosen automatically or set by the user. In

this case, the tolerance level is set automatically, which is in the order of 10^{-8}. This tolerance level may require substantially more computational time to achieve as compared to only 100 iterations set for the fmincon code. From a practical point of view, the fmincon code is recommended as it not only performs better in sparse eigenvector extraction but also allows flexible tuning of the training parameters.

There are 21 different fault scenarios included in the simulation of the Tennessee Eastman process. In this case study, the 15 known fault conditions are used for testing. The other 6 unknown fault conditions are not used in this case study. The reason is that the nature and the root-cause of these 6 fault conditions are unknown; therefore, it is difficult to assess the performance of various process monitoring techniques on these conditions. These tested fault conditions cover a range of fault types including step fault, random variation, and slow drift and sticking and are summarized in Table 9-2. The process is monitored for 960 sample intervals. All fault conditions are introduced at sample interval 160. The TE process data for these fault conditions can be downloaded at http://web.mit.edu/braatzgroup/links.html. The fault detection and false alarm rates for PCA, KPCA, KICA, RNPCA, and RNSPCA technique are presented in Table 3-5.

Table 3-5: Fault detection results of PCA, KPCA, KICA and semi-parametric PCA under 0% data contamination.

Fault Detection Rate (%) under 98% Confidence UCL												
Fault No.	PCA		KPCA		KICA		RNSPCA (Spearman)		RNSPCA (Kendall)		RNPCA (Kendall)	
	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE	T^2	SPE
1	99.25	99.75	99.25	99.75	100	99.75	99.25	99.50	99.75	99.50	99.62	99.50
2	98.75	98.50	98.75	98.50	99.00	98.62	98.75	98.50	98.75	98.62	98.75	98.50
3	7.00	1.50	6.38	1.50	1.5	**9.63**	8.62	1.13	7.00	1.63	4.88	1.25
4	1.25	3.75	1.25	4.00	3	2.88	5.12	1.50	**7.25**	1.25	1.50	3.25
5	24.25	18.88	24.12	23.25	29.75	26.12	29.50	21.50	**32.25**	19.88	24.26	18.50
6	100	100	100	100	100	100	100	100	100	100	100	100
7	52.50	25.50	52.25	26.50	47.63	37.38	50.00	35.25	51.50	32.62	**54.63**	25.87
8	97.50	96.50	97.50	97.38	98.38	98.00	97.25	98.50	97.38	97.75	97.62	97.50
9	3.00	6.63	3.00	6.00	2.13	1.63	3.00	3.75	3.62	3.00	4.62	3.75
10	53.13	31.50	53.25	33.00	**86.62**	86.12	58.13	50.62	**62.88**	45.50	55.25	31.13
11	17.25	34.75	17.00	34.50	36.25	**37.38**	25.75	35.75	**27.12**	32.87	19.13	35.00
12	98.62	82.87	98.62	87.25	99.75	98.88	98.50	98.12	98.12	97.38	98.75	83.25
13	95.63	95.00	96.63	95.10	96.00	94.87	96.00	94.63	96.00	93.75	95.63	95.13
14	97.50	100	97.25	100	97.5	100	91.25	100	85.38	100	98.12	100
15	7.00	1.50	6.88	1.13	14.50	4.25	12.00	6.00	**16.13**	2.50	9.88	3.62
False Alarm Rate (%) for IDV15 under 98% confidence UCL												
	1.87	1.87	1.25	1.87	3.75	1.25	1.87	1.86	1.25	1.86	1.25	1.25

The fault conditions in which the proposed techniques outperform the other techniques are marked in bold. For fault conditions 3, 4, 9, and 15, the closed-loop controller of the system quickly corrects the disturbance caused by the fault and brings the system back to normal operating region. This leads to low fault detection rates of all the techniques. For all the other fault conditions, the performance of KICA is superior to PCA on both statistics and is slightly better than KPCA. On the other hand, the proposed RNPCA also demonstrate superior performance to PCA. It is noted that the RNPCA techniques are not able to retain non-Gaussian features in the subspace. Nonetheless, under such a high dimensional setting, the subspace components (PCs and ICs) are obtained by summing a large amount of original process variables; these subspace components tend to have more Gaussian variation as compared to the first case study where there is only three process variables, as a result of central limit theorem. This is the main reason why the RNPCA techniques which are also able to model nonlinear correlation structures offer similar detection rates as compared to KICA. The RNSPCA techniques are observed to produce slightly better results as compared to the non-sparse versions. This could be explained by the fact putting a sparsity constraint on the eigenvectors reveals more meaningful patterns from the process data; other non-relevant information is filtered out through this process. Therefore, a better generalization capability is achieved leading to improved process monitoring performance.

It is also noticed that the proposed techniques are computationally more efficient than KPCA and KICA which adopt kernel transformation. In this case study, 960 data samples are collected to capture the behaviour of only 33 process variables. The kernel transformation is far more time consuming since there are a lot more data samples than dimensions. To further demonstrate the performance of the proposed technique in comparison to the other process monitoring techniques, the process monitoring charts of Fault 5 and Fault 15 are shown from Figure 3-14 to Figure 3-15. The monitoring windows in which the proposed techniques identify more faulty data samples are circled in red.

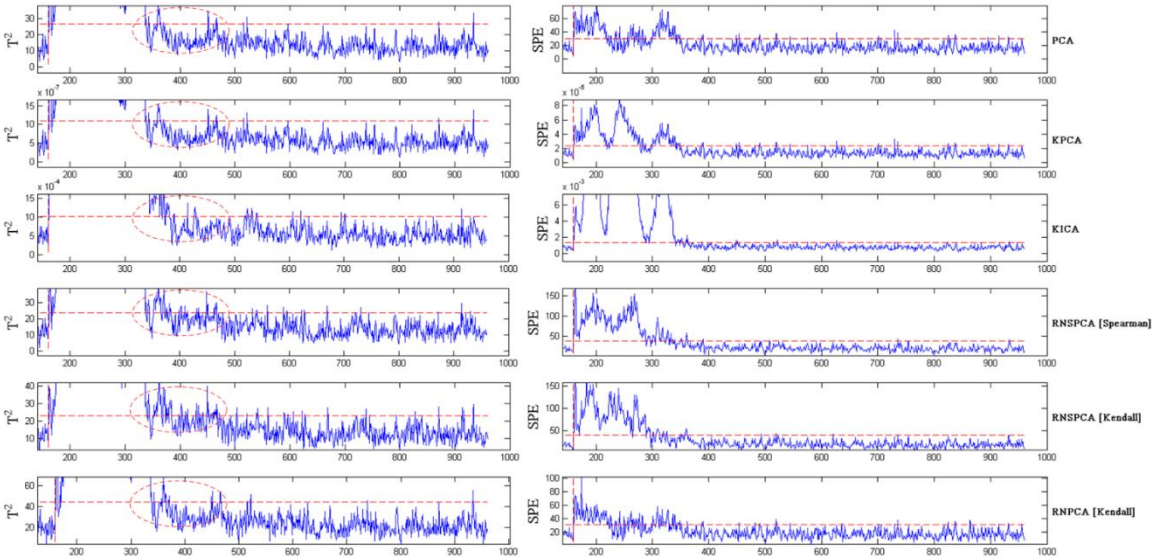

Figure 3-14: Process monitoring charts for Fault 5.

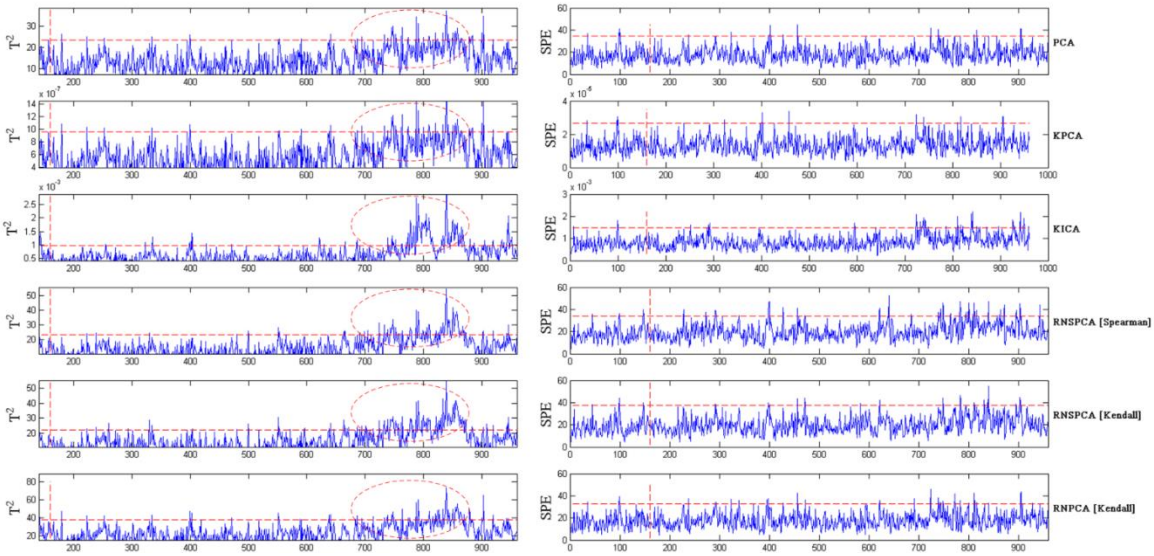

Figure 3-15: Process monitoring charts for Fault 15.

For fault diagnosis, fault condition 10 is used to test the proposed RNPCA and RNSPCA techniques. Their performances are also compared to the standard PCA. KPCA and KICA are not tested in this case as the intractable kernel transformation makes it practically not possible to compute variable contribution through reverse projection. The fault diagnosis results for the tested conditions are presented in Figure 3-16.

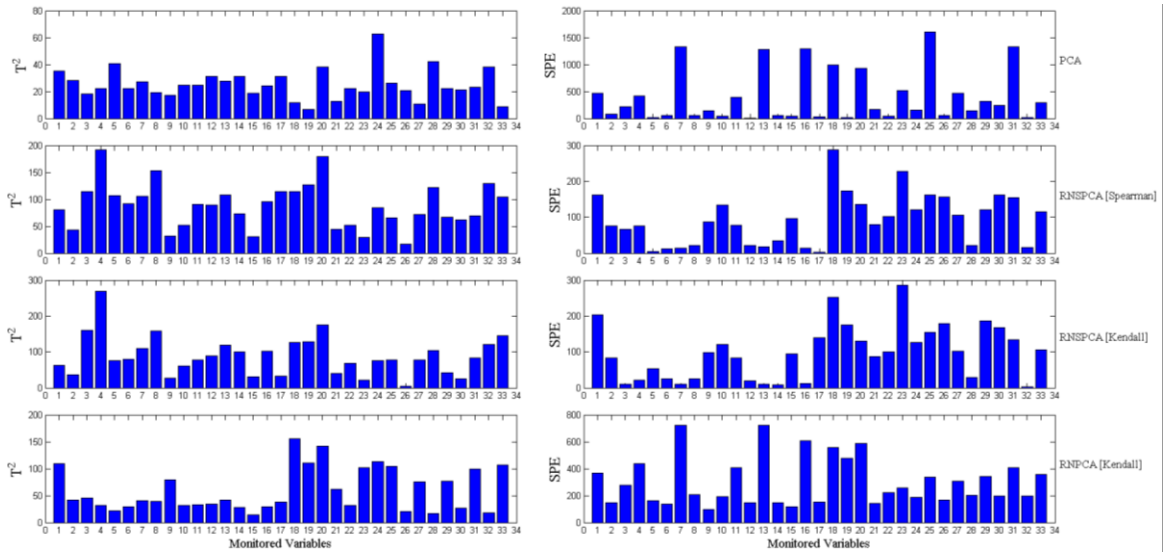

Figure 3-16: Variable contribution charts of fault 10.

Similar to the first case study, the contribution plots are generated by summing the contribution of each process variable to first 100 T^2 and SPE statistics after the introduction of fault. For fault 10, the abnormal variation in the C feed temperature (X4) directly upsets the first downstream unit—the product stripper. As a consequence, the temperature inside the product stripper (X18) is first affected. Subsequently, the pressure of the recycle flow coming out from the top of the stripper is also impacted which results in abnormal change in the compressor work (X20) for purging operation. The T^2 and SPE statistics of the RNSPCA correctly identify associated process variables. In contrast, the standard PCA fails to identify undesired behaviours of any closely related process variables. Although, T^2 statistic of RNPCA successfully located (X18), its performance is still sub-optimal comparing its sparse counterparts. Results from this more complex case study have further consolidated the viability of the proposed techniques.

In the second part of this case study, the robustness of PCA, RNPCA, and RNSPCA are tested with contaminated data. In a similar setting to the first case study, the normal process data is contaminated with a contamination rate ranging from 5% to 80%. The PDFs of the contaminated process variable X1, X10, X12 at 5% of contamination rate are shown in Figure 3-17.

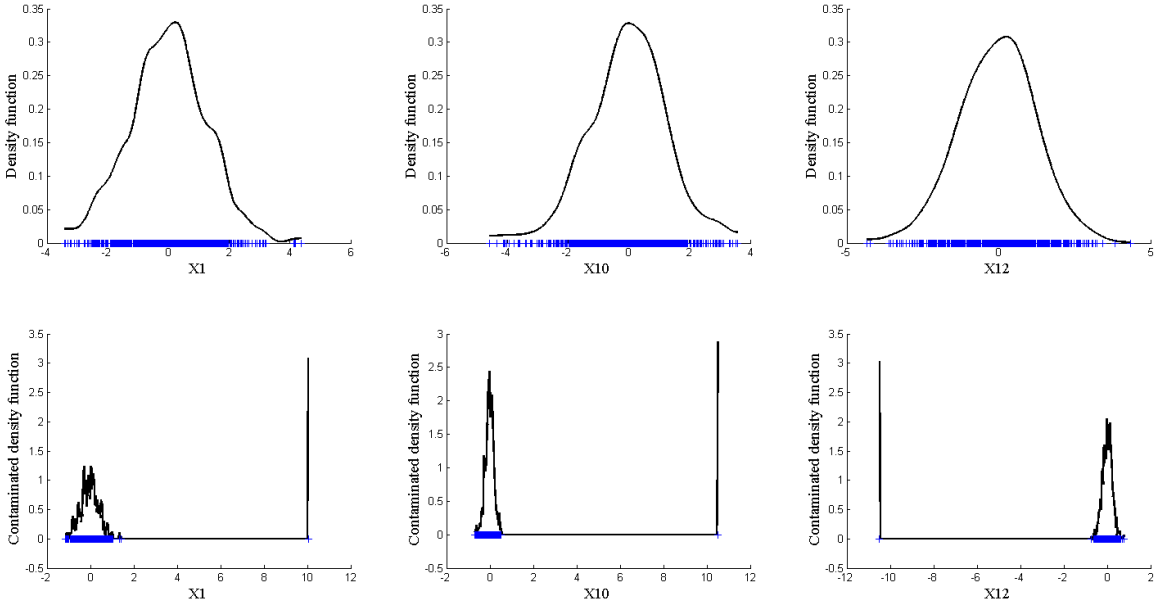

Figure 3-17: Probability density functions of the contaminated process variables at 5% contamination rate.

The error quantities including Angle, MSE and percentage of inverted correlation coefficients for all these techniques are presented in Figure 3-18.

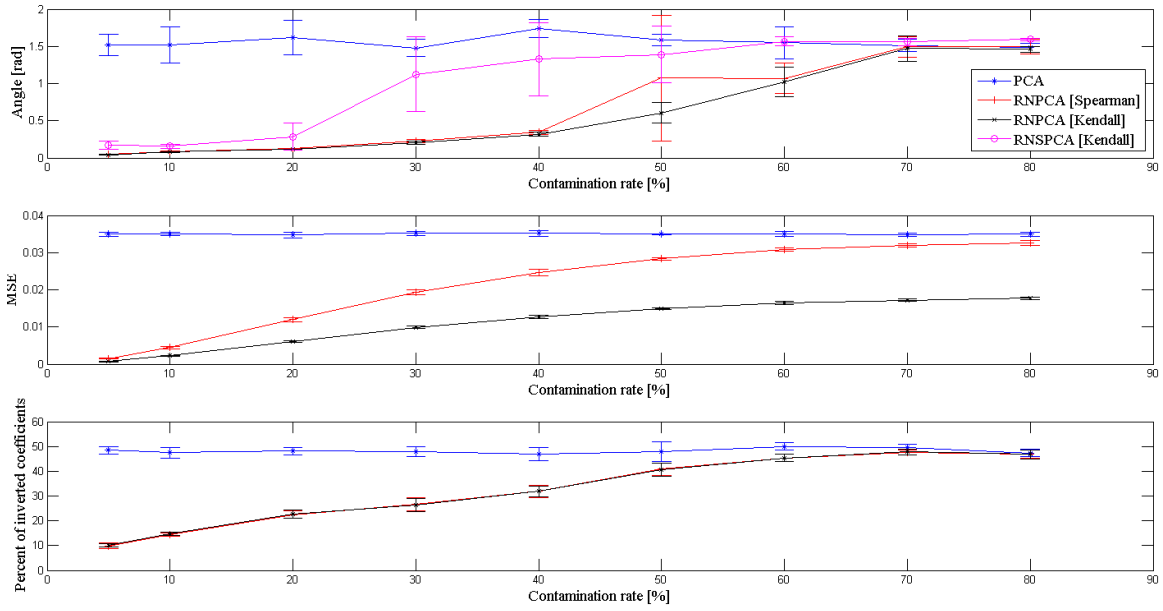

Figure 3-18: Error quantities for robustness measure of PCA, RNPCA, and RNSPCA.

It is readily seen that the Pearson's correlation measure is disrupted at a very low contamination rate (5%)—all three error quantities remain at an almost constant level throughout the contamination range. For other techniques, the ones adopting the Kendall tau's correlation measure are more robust to data contamination. Regarding the sparse feature discovery, however, the sparsity constraint enforced on the eigenvector extraction process seems to have adverse impact on the robust properties of the Kendall tau's

correlation measure. Nevertheless, RNSPCA is still more robust comparing to the standard PCA. It is therefore verified using this case study that the proposed RNSPCA offers better process monitoring performance against the conventional techniques while being more robust.

The ability to deal with data structure in which there is more process variables than the number data points ($N < d$) is another crucial element to evaluate the applicability of the proposed method. In this respect, the fault condition IDV11 is used to evaluate the performance of the RNSPCA under rank deficiency condition. The number of training data samples is reduced to 16 which is only half of the number of process variables. To cope with the rank deficiency in data, a kernel eigenvalue decomposition method proposed by Wu, et al.[64] is integrated with the proposed method. Specifically, the sequential sparse eigenvector extraction procedure is applied to the Kendall tau's correlation matrix formed by $\langle \mathbf{X}, \mathbf{X}' \rangle$ instead of $\langle \mathbf{X}', \mathbf{X} \rangle$; that is each column of $\mathbf{X}$ is considered as a data sample rather than its rows. The maximum number of pseudo-eigenvectors is equal to the number of training data samples, which is 16. However, the eigenvector matrix for online projection of KEVD is determined in a different way as following.

$$\mathbf{Q} = \left\langle \mathbf{X}', \mathbf{Q}_{KEVD} \right\rangle \mathbf{\Sigma}_{KEVD}^{-1/2} \tag{3.31}$$

where $\mathbf{Q}_{KEVD}$ is determined using the generalized deflation procedure outlined in the last part of Section 3.1, while $\mathbf{\Sigma}_{KEVD}$ is the pseudo-eigenvalue matrix whose diagonal elements are the ranked pseudo-eigenvalues of the matrix $\langle \mathbf{X}, \mathbf{X}' \rangle$. The fault detection results for PCA and RNSPCA using KEVD are presented in Figure 3-19.

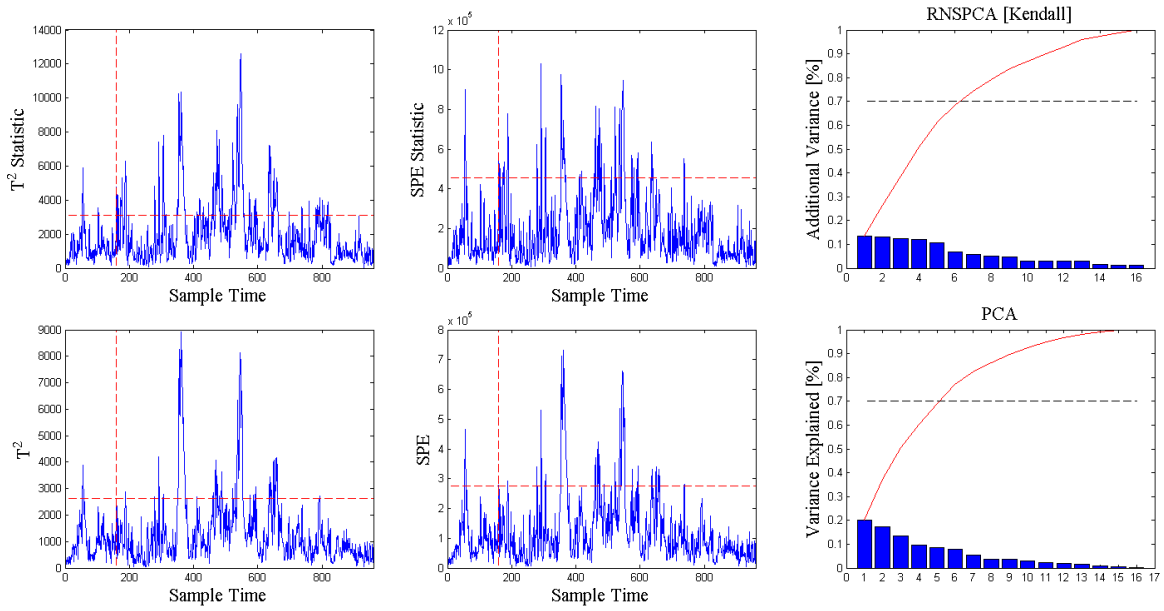

Figure 3-19: Fault detection results of RNSPCA based on KEVD.

The number of eigenvectors is set to 6 for both methods according to the scree plot cross-validation. In addition, the fault detection rate and false alarm rates of both methods

are summarized in Table 3-6. It is demonstrated here the proposed technique is more effective in handling rank deficiency through adopting KEVD as compared to conventional PCA.

Table 3-6: Fault detection rate of PCA and RNSPCA based on KEVD on IDV 11.

	Fault Detection Rate [%]		False Alarm Rate [%]	
	T^2	SPE	T^2	SPE
KEVD RNSPCA [Kendall]	20.50	9.50	3.13	2.48
KEVD PCA	11.38	8.62	1.25	1.87

3.6 Conclusions

A robust, nonlinear, and sparse PCA method (RNSPCA) is proposed in this study to address the four most prominent shortcomings of the standard PCA. To capture the nonlinear correlation structure, the Spearman's and Kendall tau's correlation matrices are used instead of the Pearson's correlation matrix. In addition, the Spearman's and Kendall tau's correlation measures are also more robust as the large deviation of the data outliers is scaled down to its rank. A set of sparse eigenvectors are then extracted from these correlation matrices. The use of sparse eigenvectors reveals meaningful pattern from the data. The RNSPCA is tested on two industrial case studies. It is demonstrated that the proposed technique can increase the amount of information retained in the normal subspace resulting in better detection of abnormal systemic variation as compared to PCA. The performance of the proposed technique is also comparable to that of the KPCA and KICA at a lower computational cost. As the data contamination rate increases, it is also shown that Kendall tau's correlation matrix is more robust as compared to Spearman's correlation matrix. This results also agree with the observations in the work of Croux, Dehon[51].

In the future work, a major disadvantage of the Spearman's correlation coefficient will be addressed; the Spearman's correlation coefficient is only able to capture the monotonic nonlinear relationship. For non-monotonic case, such as $y = x^2$, Spearman's rank correlation coefficient completely fails to identify any relationships. A variety of nonlinear dependence measures that are not restricted to monotonic relationships will be integrated with the proposed semi-parametric PCA framework for better process monitoring performance. The sparse version of the proposed method will also be applied to large-scale systems for further validation.

4 Nonlinear Gaussian Belief Network Based Fault Diagnosis for Industrial Processes

Abstract

A Nonlinear Gaussian Belief Network (NLGBN) based fault diagnosis technique is proposed for industrial processes. In this study, a three-layer NLGBN is constructed and trained to extract useful features from noisy process data. The nonlinear relationships between the process variables and the latent variables are modelled by a set of sigmoidal functions. To take into account the noisy nature of the data, model variances are also introduced to both the process variables and the latent variables. The three-layer NLGBN is first trained with normal process data using a variational Expectation and Maximization algorithm. During real-time monitoring, the online process data samples are used to update the posterior mean of the top-layer latent variable. The absolute gradient denoted as G-Index to update the posterior mean is monitored for fault detection. A multivariate contribution plot is also generated based on the G-index for fault diagnosis. The NLGBN-based technique is verified using two case studies. The results demonstrate that the proposed technique outperforms the conventional nonlinear techniques such as KPCA, KICA, SPA, and Moving Window KPCA.

Keywords: online fault diagnosis, nonlinear and noisy processes, nonlinear Gaussian belief network, PCA, KPCA, KICA, SPA, MWKPCA.

4.1 Introduction

The continuing development of process technology greatly enhances the versatility and productivity of modern industrial processes. The advanced features of industrial processes are achieved by seamless integration of functions of a large number of components. To ensure a safe and optimal operation, it is often required that the states of these components be monitored on a real-time basis. This necessitates online measurement of a large number of process variables which in turn leads to generation of a massive amount of high-dimensional data vectors. As the process is constantly subjected to systematic and external disturbances, the process data are often noisy and contain intricate information regarding the highly nonlinear interactions between process variables. This underlying nature of data has incapacitated the application of the traditional first-principle model-based process monitoring.[65] To address the presented issues, Multivariate Statistical Process Monitoring (MSPM) methods have been developed. These methods extract latent variables from the high-dimensional process data to detect and diagnose various faults.[2-4,7,33,66]

Principal Component Analysis (PCA), as a multivariate statistical analysis technique, has been extensively applied with success to process monitoring of many industrial cases.[7-9,67] In the conventional PCA algorithm, the process variables are linearly related to a smaller number of latent Gaussian variables through a set of orthogonal weight vectors. The objective of the algorithm is to determine a weight vector for each latent Gaussian variable such that the maximum possible variance in the direction of the weight vector is inherited from the process data.[68] In fact, the optimal weight vectors can be computed efficiently by performing Eigen Decomposition on the covariance matrices of the process data. As the weight vectors are orthogonal, each latent variable can only retain a limited amount of variance. These latent variables are then ordered according to the amount of the variance they retain; the one retaining the most variance is the first Principal Component (PC) and the latent variable capturing the second most variance is the second PC, and so forth.[6,69] Similar to many covariance-based models, PCA model is very sensitive to outliers presented in noisy data; this may lead to inaccurate feature extraction from noisy data.[39,70,71] For PCA-based process monitoring, the Hoteling's T^2 statistic and squared prediction error (SPE) are computed and monitored to detect process abnormalities. Multivariate contribution plots are also generated based on these two statistics to isolate the root-cause variable.[72] However, in large-scale industrial processes, the latent variables may not follow Gaussian distribution. In addition, the close integration of functionality of different units makes the interactions between monitored process variables extremely nonlinear. Furthermore, the process data vectors often contain noise due to the presence of various external disturbances. These three conditions substantially limit the capability of PCA. On the other hand, the Independent Component Analysis (ICA) does not assume Gaussian distribution of latent variables but rather forces them to be as independent (non-Gaussian) as possible.[1,16] One of the most applied ICA algorithms is the FastICA algorithm in which process data is first whitened using PCA and is then linearly projected into the latent space through a set of non-orthogonal weight vectors.[73] The optimal weight vectors are determined by iteratively maximizing the

Kurtosis (measuring non-Gaussianity) of the posteriori distribution across the latent variables. However, the Kurtosis (and many other high-order statistics) is very sensitive to outliers in noisy input data.[74,75] In addition, the use of PCA in data whitening further weakens the capability of ICA to extract useful features from noisy data. For ICA-based process monitoring, the T^2 and SPE statistics are monitored for fault detection and multivariate contribution plots are also generated for fault diagnosis.[1] Although ICA is able to retain the non-Gaussian features of the process, the nonlinear links relating process variables to the latent variables and the noisy nature of the process data are not exploited leading to sub-optimal performance. To further address the static issue of PCA and ICA, the statistical pattern analysis (SPA) technique is developed[76] by adopting the moving window approach while exploring higher order statistics. The SPA method first divides normal training data samples into non-overlapping windows. For each window, three groups of statistics are obtained: the low order statistics (mean and variance), time dependent statistics (autocorrelations and cross correlations) and the higher order statistics (kurtosis and skewness). These statistics are then regrouped to form a new training data sample for the SPA method. The standard PCA is then conducted on the newly training data samples to obtain a dynamic PCA model capable of extracting non-Gaussian features. A major disadvantage of such a method compared to the proposed method is attributed to the use of third order kurtosis and fourth order skewness which are extremely sensitive to data outliers. Additionally, if the window size is chosen to be small due to the limited number of training data samples, the number of variables in the new data matrix might be significantly larger than the number of samples. The smaller amount of data samples might not contain enough information for robust feature extraction.

Kernel PCA and Kernel ICA are proposed to relax the limitation of linear projection of PCA and ICA, respectively. These two techniques utilize Kernel tricks to map the process data vector into a high-dimensional (polynomial Kernel) or infinite-dimensional (radial basis Kernel) Kernel space with the hope that process variables become linearly related to the latent variables in high dimension.[77-80] In the case of KPCA, standard Singular Value Decomposition (SVD) can be performed on the mapped process data matrix in the Kernel space to extract PCs. The T^2 and SPE statistics can be computed and monitored in the similar way to PCA for fault detection.[20] KPCA can effectively deal with process nonlinearity; however, since PCA still plays a key part in this technique, KPCA is not able to extract non-Gaussian features from the noisy process data either. With regards to KICA, KPCA are first used to perform Kernel whitening and centring on the mapped process data. Subsequently, the iterative ICA algorithm is used to extract ICs in the Kernel space.[18] Process monitoring based on KICA is very similar to that of KPCA. KICA addresses the issue of nonlinear and non-Gaussian features of large-scale processes. Nevertheless, due to the usage of KPCA whitening and Kurtosis for feature extraction, KICA process monitoring may yield unsatisfactory results on noisy data. To take into account the autocorrelations and cross correlations of process data samples, the Moving Window KPCA (MWKPCA) is proposed.[81] A KPCA model is constructed for every monitoring window containing a predefined number of samples. Each window is also highly overlapping—a new monitoring window is obtained by discarding an old sample and including a new sample. Subsequently, the monitoring statistics (T^2 and SPE)

of an online data sample is computed with respect to a historical KPCA model obtained several steps earlier. Since a KPCA model is computed for each new window, the MWKPCA might be computationally very expensive. Moreover, the online fault data samples are also grouped in windows for building the KPCA models. This can seriously degrade the fault detection performance if the system is stabilized after fault; if such condition arises, the KPCA model built for each window of fault samples do not differ significantly from each other resulting in low fault detection rate. Another major drawback of Kernel methods is the intractable Kernel mapping which makes it practically impossible to determine individual contribution of each process variable through reverse projection. Therefore, fault diagnosis techniques based on KPCA, KICA and MWKPCA are yet to be studied.[18,20]

Nonlinear Gaussian belief networks (NLGBN) are a type of generative models capable of extracting nonlinear Gaussian latent variables from noisy data.[11] The process variables are nonlinearly related to the latent variables through a set of nonlinear functions. In addition, model variances are also assigned to both process variables and latent variables to take into account the noisy nature of the process data. The outputs of the network are the posterior means and variances of the latent variables. These inferential statistics represent the low dimensional feature of the Process data. They are estimated by iteratively minimizing the Kullback-Leibler divergence between the variational distribution and the true posterior distribution of the latent variables.[82] In this study, a NLGBN-based online fault detection and diagnosis technique is proposed for industrial processes. Firstly, a three-layer NLGBN is trained with normal operational data samples of the process by using an efficient variational expectation maximization (EM) algorithm developed by Frey, Hinton[82]. The training continues until the gradient to update the posterior mean of the top-layer latent variable stabilizes at a value close to zero indicating the convergence of posterior mean. The model variance and the nonlinear projection weights of each process variable are also determined. Subsequently, during real-time monitoring, the model variance and the nonlinear weights are fixed while the online process data samples are used to update the posterior mean of the top-layer latent variable again. The gradient for updating this posterior mean is used as the monitoring index (G-index) for online fault detection. When the process is operating normally, the data pattern is rather similar to that of the training data resulting in almost no change to the index. If a fault is introduced into the system, the process data pattern starts to diverge from normal leading to a significant deviation of the G-index as the algorithm strives to adapt the NLGBN model to a new problem space. The control limit for the G-index is estimated by adopting a nonparametric Kernel density estimator.[62] For fault diagnosis, the individual contribution of each process variable can be determined by calculating its contribution to the G-index. It is noted that the latent variables of NLGBN are still assumed to follow Gaussian distribution. This assumption is necessary to obtain a tractable cost function to estimate the true posterior distribution. Despite this fact, it is shown in the case studies that the latent variation of the process may consist of both Gaussian and non-Gaussian features. The ability to extract non-Gaussian features may not be the only determinant factor for fault diagnosis techniques to achieve high performance in real-time process monitoring. On the other hand, the proposed technique which effectively incorporates the

effect of noise in process data is demonstrated to outperform KICA and KPCA though only the Gaussian features of the latent variation are retained.

The reminder of the article is organized into four sections. A brief review of the fundamental principle of the NLGBN is presented in the Background section. The derivation of the proposed NLGB-based fault diagnosis technique is illustrated in the Methodology section. In the Case Studies section, the efficacy of the proposed technique is first validated on a simple nonlinear numerical model and is then verified by the well-established Tennessee Eastman chemical process simulation. The superiority in performance of the proposed technique to the conventional techniques is demonstrated by comparing the results. The major conclusions are summarized in the final section.

4.2 Background

Nonlinear Gaussian Belief Networks (NLGBN) was first developed as a generative model for vision and speech recognition.[11] NLGBN assumes the noisy input signal can be explained by a set of latent nonlinear Gaussian units. A typical graphical representation of a two-layer NLGBN is presented in Figure 4-1(a). The noisy input signals/process variables $x_i, i \in \{1:n\}$ having model variance δ_i^2 are nonlinearly related to the latent unit $y_j, j \in \{1:p\}$ with a model variance v_j^2 through a set of weights w_{ji} and nonlinear functions $f_j(y_j)$. For a robust feature extraction, it is essential to introduce model variances at the latent variable layer as the noise of input data still exists after nonlinear transformation. In contrast, the generative model for PCA/ICA is shown in Figure 4-1(b) in which the process variables are assumed to be noise-free and linearly related to the latent variables. This configuration makes it easy to solve for the optimal weights using SVD for PCA and also it allows a closed-form posterior distribution across latent variables to be acquired for ICA algorithm.

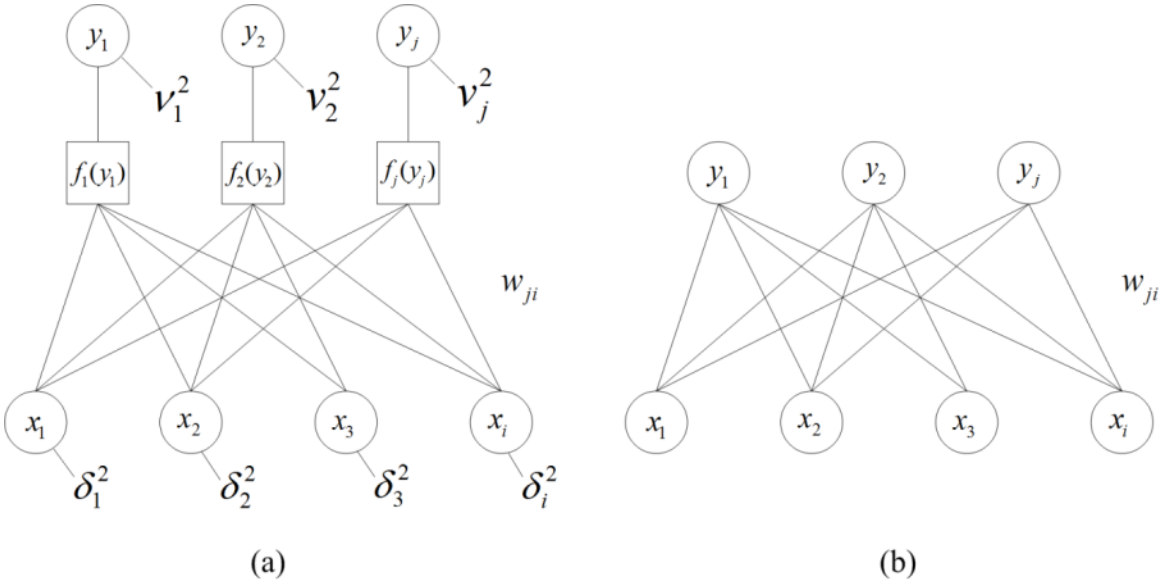

Figure 4-1: Generative models of NLGBN (a) and PCA/ICA (b)

In NLGBN, the latent variables and process variables are both assumed to have Gaussian distribution under the influence of the model variance and the nonlinear functions have tractable derivatives. These two conditions are sufficient for the variational EM learning to estimate the inferential statistics.[82] The generative distribution of the NLGBN can be obtained as follow.

$$\begin{aligned} p(\mathbf{x} \cap \mathbf{y}) &= \prod_{j=1}^{p} p(y_j) \prod_{i=1}^{n} p\left(x_i \mid \{y_j\}\right) \\ &= \prod_{j=1}^{p} \frac{1}{v_j} \phi\left(\frac{y_j - E[y_j]}{v_j}\right) \prod_{i=1}^{n} \frac{1}{\delta_i} \phi\left(\frac{x_i - \sum_j w_{ji} f_j(y_j)}{\delta_i}\right) \end{aligned} \tag{4.1}$$

where $\mathbf{x} = \{x_i\}$, $\mathbf{y} = \{y_j\}$, $\phi(\cdot)$ is the probability density function of the standard normal distribution and $E(y_j)$ is the expectation of $y_j$under model variance v_j^2. Since both the latent variables and process variables follow Gaussian distribution, the posterior probability distribution over the latent variables should also follow Gaussian distribution due to convolution. Therefore, it is possible to define the initial estimate of the posterior distribution explicitly.

$$q(\mathbf{y}) = \prod_{j=1}^{p} \frac{1}{\sigma_j} \phi\left(\frac{y_j - \mu_j}{\sigma_j}\right) \tag{4.2}$$

where $q(\cdot)$ is also known as the variational distribution and μ_j and σ_j are the variational mean and standard deviation.[82] The objective of the variational EM learning algorithm is to iteratively minimize the Kullback-Leibler divergence between the variational

distribution and the true posterior distribution over **y**. In this case, the following bounded objective function proposed by [83] is adopted.

$$F = E[\log p(\mathbf{x} \cap \mathbf{y})] - E[\log q(\mathbf{y})] \leq \log p(\mathbf{x}) \tag{4.3}$$

where $E[\cdot]$ is the expectation under $q(\mathbf{y})$. It can be shown that when $q(\mathbf{y}) = p(\mathbf{y}|\mathbf{x})$, that is the variational distribution is identical to the posterior distribution (Kullback-Leibler divergence = 0), the objective function $F = \log p(\mathbf{x})$ is maximized meaning that the bound in Eq.(4.3) is tight. The proof of this condition is presented in the Section 9.7 of the Appendices. Once the true posterior distribution is obtained, the inferential statistics (μ_j and σ_j) represent the low dimensional feature of the noisy process data.

4.3 NLGBN-based online fault diagnosis

In this study, a three-layer NLGBN is constructed and trained to conduct online fault diagnosis on nonlinear and noisy processes. The bottom-layer units of the NLGBN are visible units each of which represents one process variable. A set of sigmoidal functions and weights are added between the bottom layer and middle layer to achieve nonlinear projections. Furthermore, the middle-layer latent units are linearly connected to one top-layer latent unit such that a single set of inferential statistics can be generated at the top layer. Each layer of units has a different set of model variances. The advantages of this setup are:

- It represents the simplest generative model with adequate complexity to capture the nonlinear relationships between the process variables and latent variables;
- As there is only one top-layer latent variable, it is possible to detect and diagnose fault by only monitoring a single parameter which is the absolute gradient to update the posterior mean of the top-layer latent unit. This gradient is denoted as G-index in the context of this study.

The gradient for the posterior variance is not monitored as it only describes the spread (noise level) of the data but not the moving trend during online monitoring. The graphic structure of the three-layer NLGBN is shown in Figure 4-2.

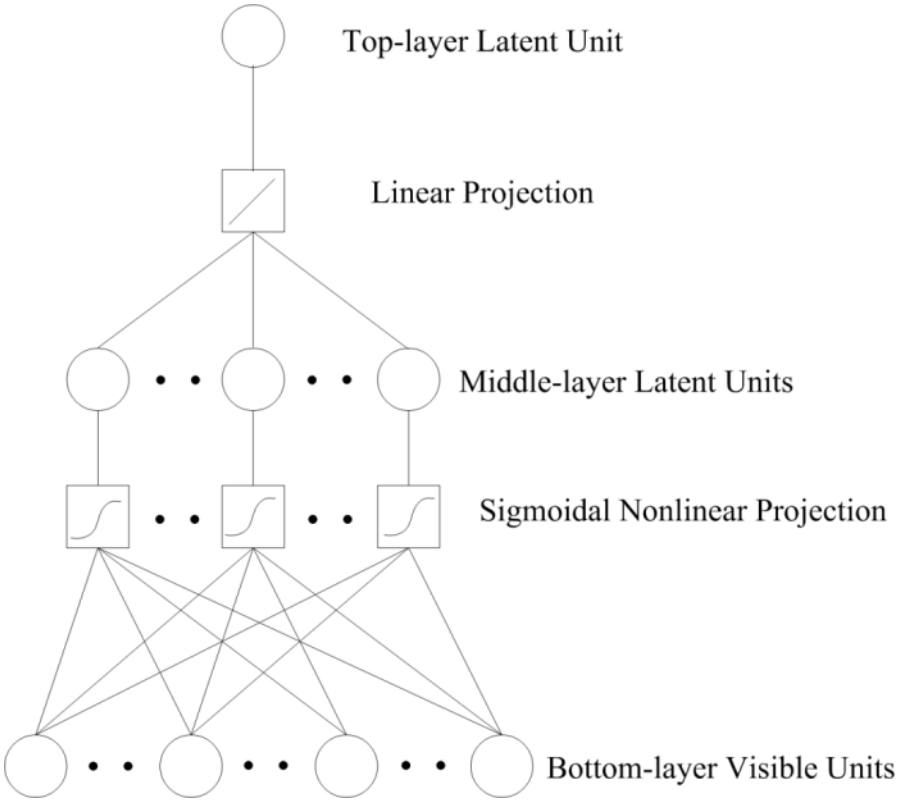

Figure 4-2: Three-layer NLGBN structure

To attain a set of concise mathematical expressions of the three-layer NLGBN, the units in the network are classified into parental units and child units only. The notation for the set of parental units is $\mathbf{x}_j = \{x_j\}_{j\in A_i}$and the notation for a set of child units is $\mathbf{x}_i = \{x_i\}_{i\in C_j}$, where A_i represents the set of indices for the parents of unit x_i while C_j includes the set of indices for the children of unit x_j. A three-layer NLGBN with a total number of N units has a combined set of indices $A_i \cup C_j = \{1{:}\,N\}$. Expression (4.1) can be re-written as follow.

$$p\left(\mathbf{x}_i \cap \mathbf{x}_j\right) = \prod_{i=1}^{N} p\left(x_i \mid \left\{x_j\right\}_{j\in A_i}\right) = \prod_{i=1}^{N} \frac{1}{\varepsilon_i}\phi\left(\frac{x_i - \sum_{j\in A_i} w_{ji} f_j(x_j)}{\varepsilon_i}\right) \tag{4.4}$$

where ε_i is the set of standard deviations for all the units in the network. The expectation over the log likelihood of the generative distribution with respect to the variational distribution $q(x_j)$ is expressed as.

$$E\left[\log p\left(\mathbf{x}_i \cap \mathbf{x}_j\right)\right] = E\left[\sum_{i=1}^{N} \log \frac{1}{\varepsilon_i}\phi\left(\frac{x_i - \sum_{j\in A_i} w_{ji} f_j(x_j)}{\varepsilon_i}\right)\right]$$
$$= -\sum_{i=1}^{N} \frac{\log\left(2\pi\varepsilon_i^2\right)}{2} - \sum_{i=1}^{N} \frac{E\left[\left(x_i - \sum_{j\in A_i} w_{ji} f_j(x_j)\right)^2\right]}{2\varepsilon_i^2} \tag{4.5}$$

For a projection function $f(x_j)$, $E[f(x_j)]$ is defined as.

$$E[f(x_j)] = \int_{x_j} f(x_j) q(x_j) dx_j$$
$$q(x_j) = \frac{1}{\sigma_j}\phi\left(\frac{x_j - \mu_j}{\sigma_j}\right) \tag{4.6}$$

where, μ_j and σ_j are the variational mean and variance, respectively. To get rid of $E[\cdot]$, the mean output of unit *j* and variance at the output of unit *j* are first determined.

$$M_j(\mu_j, \sigma_j) = \int_{x_j} f_j(x_j) q(x_j) dx_j = E\left[f_j(x_j)\right]$$
$$V_j(\mu_j, \sigma_j) = \int_{x_j} \left[f_j(x_j) - M_j(\mu_j, \sigma_j)\right]^2 q(x_j) dx_j = E\left[\left[f_j(x_j) - M_j(\mu_j, \sigma_j)\right]^2\right] \tag{4.7}$$

where $M_j(\mu_j, \sigma_j)$ is the mean output and $V_j(\mu_j, \sigma_j)$ is the variance at the output of unit *j*. Next, add and subtract both μ_i and $\sum_{i=1}^{N} w_{ji} M_j(\mu_j, \sigma_j)$ in the term under the square $\left(x_i - \sum_{j\in A_i} w_{ji} f_j(x_j)\right)^2$ giving.

$$E\left[\left(x_i - \sum_{j\in A_i} w_{ji} f_j(x_j)\right)^2\right]$$
$$= E\left[\left(\overbrace{\left(x_i - \mu_i\right)}^{1} + \overbrace{\left(\mu_i - \sum_{j\in A_i} w_{ji} M_j(\mu_j, \sigma_j)\right)}^{2} + \overbrace{\sum_{j\in A_i} w_{ji}\left(M_j(\mu_j, \sigma_j) - f_j(x_j)\right)}^{3}\right)^2\right] \tag{4.8}$$

As a result, this term is re-organized into three terms as shown in Eq.(4.8). Term 1 becomes zero under the expectation. The cross-terms formed by multiplying term 2 by

term 3 due to the square operation also disappear under the expectation.[§] Finally, Expression (4.5) simplifies as.

$$E\left[\log p\left(\mathbf{x}_i \cap \mathbf{x}_j\right)\right] = -\sum_{i=1}^{N} \frac{\log\left(2\pi\varepsilon_i^2\right)}{2} - \sum_{i=1}^{N} \frac{\left[\mu_i - \sum_{j\in A_i} w_{ji} M_j(\mu_j,\sigma_j)\right]^2 + \sum_{j\in A_i} w_{ji}^2 V_j(\mu_j,\sigma_j)}{2\varepsilon_i^2} \tag{4.9}$$

Likewise, the bounded objective function F in (4.3) can be written in the following concise form, as;

$$F = -\sum_{i=1}^{N} \frac{\log\left(2\pi\varepsilon_i^2\right)}{2} - \sum_{i=1}^{N} \frac{\overbrace{\left[\mu_i - \sum_{j\in A_i} w_{ji} M_j(\mu_j,\sigma_j)\right]^2}^{1} + \sum_{j\in A_i} w_{ji}^2 V_j(\mu_j,\sigma_j)}{2\varepsilon_i^2} + \sum_{j\in A_i} \frac{\left(1+\log 2\pi\sigma_j^2 - \frac{\sigma_j^2}{\varepsilon_j^2}\right)}{2} \tag{4.10}$$

where,

$$E\left[\log p\left(\mathbf{x}_j\right)\right] = E\left[\sum_{j\in A_i} \log \frac{1}{\varepsilon_j} \phi\left(\frac{x_j - \mu_j}{\varepsilon_j}\right)\right] = \sum_{j\in A_i} \frac{\left(1+\log 2\pi\sigma_j^2 - \frac{\sigma_j^2}{\varepsilon_j^2}\right)}{2} \tag{4.11}$$

It is noticed that a dummy variable μ_i is also created for the visible process variable unit at the bottom layer. In the process of learning, when a new training data sample x_i^* is available at the bottom layer, set $\mu_i = x_i^*$ and the rest of the variational statistics (μ_j and σ_j) and model parameters (ε_i and w_{ji}) are determined using the variational EM algorithm. Additionally, in Eq.(4.10), term 1 calculates the mean squared error between μ_i and summed input to unit i from its parental units j under $q(\cdot)$. This error is then down-weighted by the model variance ε_i^2. This mechanism efficiently deals with the effect of data noise in such a way that the large mean squared error generated by an outlier has

[§] To see how the cross-terms disappear, expand the product of term 2 and term 3 giving $E\left[\mu_i\left(\sum_{j\in A_i} w_{ji} M_j(\mu_j,\sigma_j) - \sum_{j\in A_i} w_{ji} f_j(x_j)\right)\right] + E\left[\sum_{j\in A_i} w_{ji} M_j(\mu_j,\sigma_j)\left(\sum_{j\in A_i} w_{ji} M_j(\mu_j,\sigma_j) - \sum_{j\in A_i} w_{ji} f_j(x_j)\right)\right]$. As $E\left[\sum_{j\in A_i} w_{ji} f_j(x_j)\right] = \sum_{j\in A_i} w_{ji} M_j(\mu_j,\sigma_j)$ (refer expression(4.7)), all the cross-terms cancel out.

small influence on the F under a large model variance. This leads to better feature extraction from noisy process data.

The variational learning of the three-layer NLGBN consists of two major steps: Expectation (E-step) and Maximization (M-step).[82] In the E-step, the model parameters ε_i and w_{ji} are randomized and the bounded objective function (4.10) is maximized with respect to the variational parameters μ_j and σ_j. The steepest gradient descent method is used for the optimization which requires the computing of the derivatives of F with respect to μ_j and σ_j, correspondingly. The following parameters are introduced to further simplify the notations.

$$\begin{aligned} \alpha_j &= M_j(\mu_j, \sigma_j) \\ \beta_j &= V_j(\mu_j, \sigma_j) \\ \gamma_j &= \sum_{j \in A_i} w_{ji} \alpha_j \end{aligned} \tag{4.12}$$

Substitute (4.12) into (4.10), F is simplified as.

$$F = -\sum_{i=1}^{N} \frac{\log\left(2\pi\varepsilon_i^2\right)}{2} - \sum_{i=1}^{N} \frac{\left[x_i^* - \gamma_j\right]^2 + \sum_{j \in A_i} w_{ji}^2 \beta_j}{2\varepsilon_i^2} + \sum_{j \in A_i} \frac{\left(1 + \log 2\pi\sigma_j^2 - \frac{\sigma_j^2}{\varepsilon_j^2}\right)}{2} \tag{4.13}$$

Subsequently, the derivatives of F with respect to μ_j and σ_j are computed as.

$$\frac{\partial F}{\partial \mu_j} = \frac{\gamma_j - \mu_j}{\varepsilon_j^2} - \frac{\partial \alpha_j}{\partial \mu_j} \sum_{i \in C_j} \frac{w_{ji}(\gamma_j - x_i^*)}{\varepsilon_i^2} - \frac{\partial \beta_j}{\partial \mu_j} \sum_{i \in C_j} \frac{w_{ji}^2}{2\varepsilon_i^2} \tag{4.14}$$

$$\frac{\partial F}{\partial \log \sigma_j^2} = -\frac{\partial \beta_j}{\partial \log \sigma_j^2} \sum_{i \in C_j} \frac{w_{ji}^2}{2\varepsilon_i^2} - \frac{\partial \alpha_j}{\partial \log \sigma_j^2} \sum_{i \in C_j} \frac{w_{ji}(\gamma_j - x_i^*)}{\varepsilon_i^2} - \frac{\sigma_j^2}{2\varepsilon_j^2} + \frac{1}{2} \tag{4.15}$$

It may be seen that $\log\sigma_j^2$ is used instead of σ_j^2. This is due to the fact that $\log\sigma_j^2$ can have negative value which allows the gradient to change direction in the problem space. The derivatives of α_j and β_j with respect to $\mu_j$and $\log\sigma_j^2$ are provided in the Section 9.8 Appendices. The initial values of the variational means (μ_j) are set to 1 for the middle layer units and the top layer unit. For the visible layer unit, $\mu_i = x_i^*$, $\sigma_j = 0$ because data samples are observed without uncertainty. The initial values for the variational variances of the middle layer units and top layer unit is set to 0.01.

For each training step $t, t \in \{1{:}T\}$, a set of variational parameters μ_j^t, σ_j^t together with α_j^t, β_j^t, γ_j^t are generated. To achieve a fast convergence, these parameters are used to initiate the next E-step.

In the M-step, it can be shown in expression (4.13) that the weights w_{ji} are in fact independent of the model variance ε_j^2; the optimal value of these two model parameters

can be obtained independently. In the case of w_{ji}, the set of weights for each unit is also decoupled from each other. These two conditions allow the optimal weights to be solved efficiently using SVD. To solve for w_{ji}, the following sufficient statistics are first computed for the current training sample x^t.[82]

$$\begin{aligned} a_{jk} &= \frac{1}{T}\sum_t \alpha_j^t \alpha_k^t, \;\; b_j = \frac{1}{T}\sum_t \beta_j^t, \;\; \mathrm{c}_{ij} = \frac{1}{T}\sum_t x_i^t \alpha_j^t, \\ d_j &= \frac{1}{T}\sum_t (\gamma_j^t - \mu_j^t)^2, \;\; e_j = \frac{1}{T}\sum_t \sigma_j^{t\,2} \end{aligned} \tag{4.16}$$

where $j = k \in A_i$. To save computational time, these statistics are accumulated for each training iteration. They are used to construct the following linear system of equations for solving w_{ji}.

$$\sum_{k \in A_i} a_{jk} w_{ki} + b_j w_{ji} = c_{ij} \tag{4.17}$$

where i is fixed for this system of equations such that w_{ji} can be solved for unit i. Expression (4.17) can be written in matrix form as shown in (4.18) which can be solved by using SVD.

$$\left[\mathbf{A}_{jk} + \mathrm{diag}(b_j)\right]\mathbf{W}_{ji} = \mathbf{C}_{ji} \tag{4.18}$$

where for each unit i, $\mathbf{A}_{jk} \in \Re^{j \times k}$ and $\mathbf{W}_{ji}, \mathbf{C}_{ji} \in \Re^{j \times 1}$. Subsequently, the model variance can also be calculated as.

$$\varepsilon_j^2 = d_j + e_j + \sum_{k \in A_j} w_{jk}^2 b_k \tag{4.19}$$

The derivation of Eq.(4.17) and Eq.(4.19) are presented in Section 9.9 Appendices. The training stops when the gradient to update the posterior mean $\frac{\partial F}{\partial \mu_j}$ of the top-layer unit converges to a value close to zero. During online monitoring, the model parameters are fixed and set $\mu_i = x_i^*$ for each new training sample. Then, the absolute value of the gradient to update the posterior mean of the top-layer latent variable $\left|\frac{\partial F}{\partial \mu_j}\right|$ which is called the G-index is monitored to determine whether there is a fault in the process. The rationale behind this monitoring technique is that since G-index has converged on normal data, any fault that introduces changes in the monitored data will force the absolute gradient to increase as the learning algorithm strives to adapt the NLGBN to a new optimal point. A major advantage of this monitoring technique is that the gradient tracks the local curvature of the highly nonlinear problem space formed by the process data which makes it sensitive to the subtle corruption of the problem space when there is a

fault. This leads to high fault detection efficiency. The control limit for the G-index is estimated using a nonparametric Kernel density estimator under 95% confidence.

$$\hat{f}_h(G_t) = \frac{1}{nh}\sum_{t=1}^{n}\phi\left(\frac{G_t - E[G_t]}{h}\right) \tag{4.20}$$

$$\int_{-\infty}^{UCL} \hat{f}_h(G_t)dG_t = 0.95 \tag{4.21}$$

where $G_t$is the G-index for the t^{th} training data sample, h is the bandwidth and ϕ is the Gaussian density function.[84] The optimal bandwidth is determined by adopting a diffusion-based plug-in selection method.[62] The upper control limit is denoted as *UCL*. The conceptual representation of the NLGBN-based fault detection technique is presented in Figure 4-3.

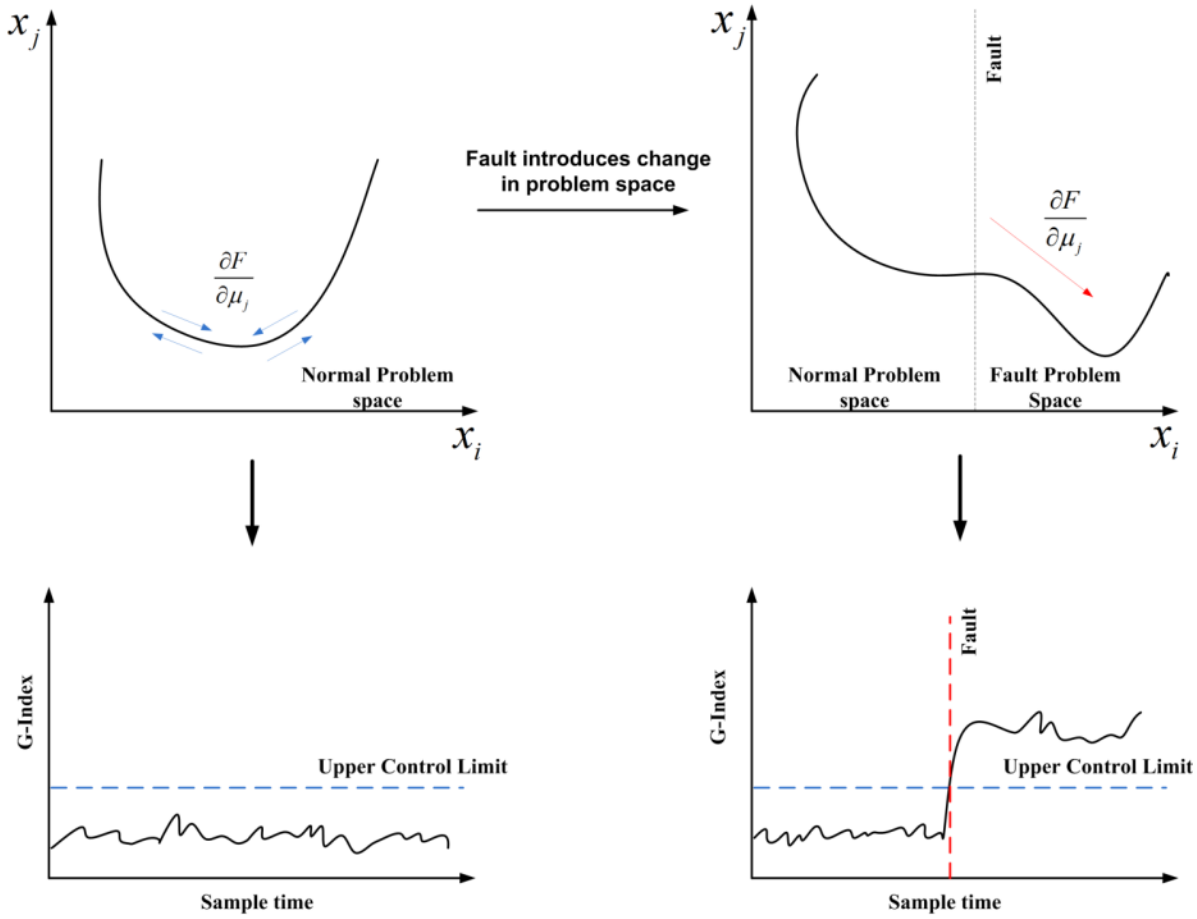

Figure 4-3: Conceptual representation of NLGBN-based fault detection

For fault diagnosis, the individual contribution of each process variable to the G-index can be easily decomposed based on Eq. (4.14). The contribution of process variable x_i to the gradient is given as.

$$cont_i\left(\left|\frac{\partial F}{\partial \mu_j}\right|\right)=\left|\frac{\gamma_j-\mu_j}{\varepsilon_j^2}-\frac{\partial \alpha_j}{\partial \mu_j}\frac{w_{ji}(\gamma_j-x_i^*)}{\varepsilon_i^2}-\frac{\partial \beta_j}{\partial \mu_j}\frac{w_{ji}^2}{2\varepsilon_i^2}\right| \tag{4.22}$$

Notice that the summation signs have disappeared and μ_i is replaced with x_i^*. This treatment allows the computation of gradient considering only the measured value of process variable x_i^*, i.e. the instant contribution of process variable x_i to the change of gradient. As compared to the traditional T^2 and SPE statistics which are computed based on noise free and relatively static feature space, the proposed contribution index is adaptive to capture the dynamics of the process, especially in fault condition, making it more sensitive in fault diagnosis. This decomposition eliminates the need of reverse projection as required by KICA/KPCA for fault diagnosis. The logical flow diagram of the NLGBN-based fault diagnosis technique is shown in Figure 4-4.

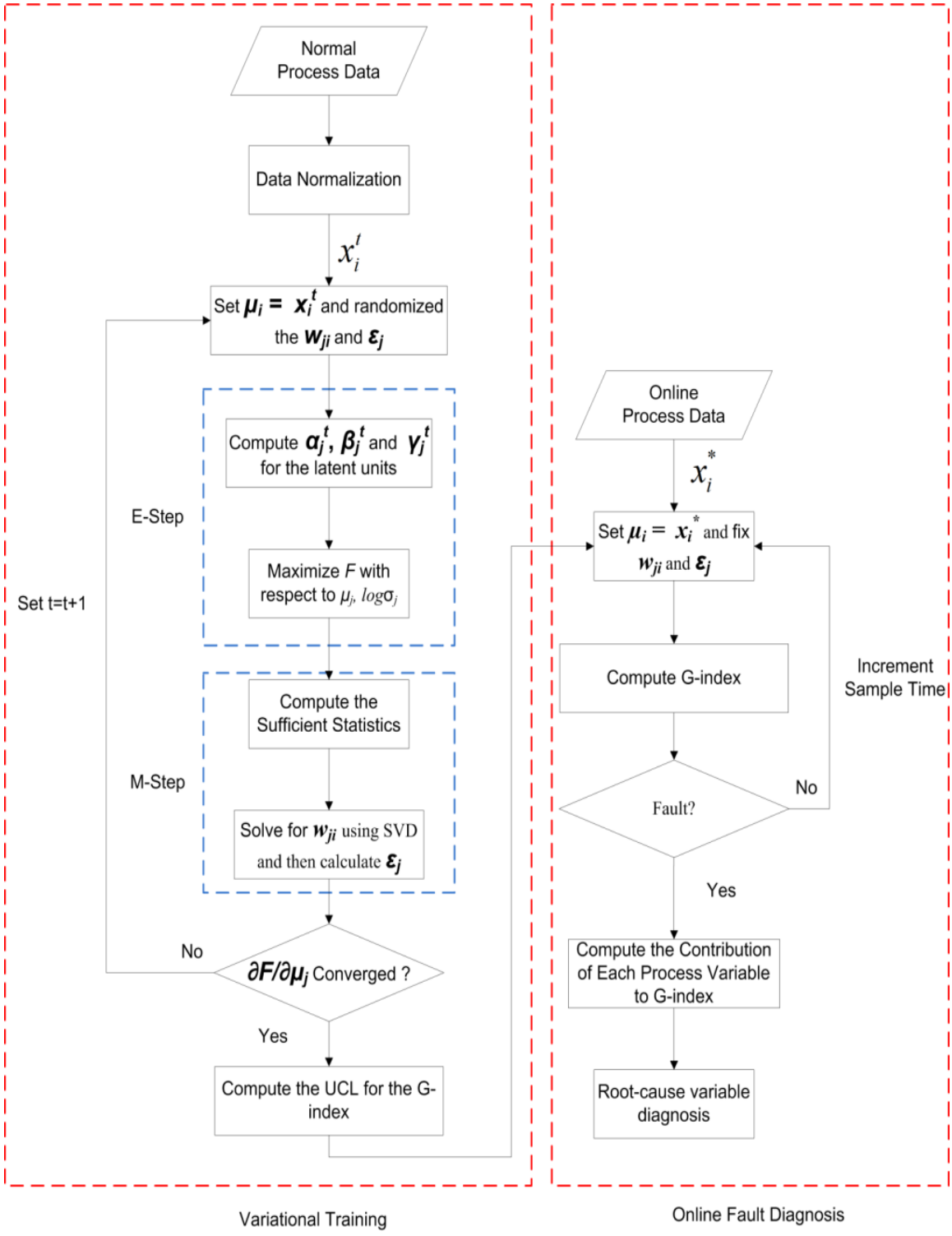

Figure 4-4: Logic flow chart of the proposed NLGBN-based fault diagnosis technique

The computational complexity of the proposed technique is also significantly lower than KICA and KPCA which perform SVD in the high-dimensional Kernel space. Consider N samples of d-dimensional data vectors are used to train the KICA/KPCA, the computational complexity in terms of big O notation for the these two techniques is $O(N^2d + N^3)$ [78,85], whereas for the proposed technique the computational complexity is

only $O[2N(d^2+d^3)]$; for every iteration, in the M-step, a SVD is performed for each latent layer (2 latent layers in total) to obtain w_{ji} for at most d process variables $O(d^3)$ [**] and computing the sufficient statistics scales linearly in $O(d^2)$.

4.4 Case Studies

4.4.1 A non-linear numerical example

The effectiveness of the proposed fault diagnosis technique is first demonstrated on a simple multivariate nonlinear process. This numerical process has four output variables and one Gaussian input variable, which are monitored to determine the state of the process. The mathematical model of this nonlinear process is shown as following.

$$\mathbf{X} = \mathbf{AZ} + \mathbf{\Phi} \tag{4.23}$$

Where $\mathbf{A} \in \mathbb{R}^{4\times 2}$ is the coefficient matrix and $\mathbf{\Phi}$ is the zero-meaned multivariate Gaussian noise having a correlation matrix of $0.25\mathbf{I}$, $\mathbf{I} \in \mathbb{R}^{4\times 4}$.

$$\mathbf{A} = \begin{bmatrix} 1 & 4 \\ 2 & 1 \\ 1 & 3 \\ 3 & 1 \end{bmatrix} \tag{4.24}$$

The input vector $\mathbf{Z}$ comprises of two signals which are generated according to the following model.

$$\mathbf{Z} = \begin{bmatrix} u^4 + 4u^3 + 2u^2 + u \\ 2u^4 + u^3 - 2u^2 + 3u \end{bmatrix} \tag{4.25}$$

Where u is the input variable and is sampled from $u \sim N(3, 0.1)$. Due to the nonlinear transformation, $\mathbf{Z}$ may have non-Gaussian variation and consequently the output vector $\mathbf{X}$ may also contain non-Gaussian features. Table 4-1 summarizes the Kurtosis[††] of the monitored variables.

Table 4-1: Kurtoses and distribution class of the monitored variables

	x_1	x_2	x_3	x_4	u
Kurtosis	-0.000161	-0.000611	0.000374	-0.000262	0
Class	Sub-Gaussian	Sub-Gaussian	Super-Gaussian	Sub-Gaussian	Gaussian

[**] This is the worst case scenario in that there are d latent variables in each latent layer. In fact, there is only one latent variable in the top latent layer in the proposed three-layer NLGBN structure.

[††] As non-Gaussian variables may not have a closed-form distribution, the expectation in the Kurtosis is replaced by sample mean in this case.

It is observed that after the nonlinear transformation, the Kurtosis of each monitored process variable is still very close to zero meaning that the shape of their distributions are very similar to that of a Gaussian distribution. From a generative model point of view, process variable distribution is generated by either linearly or nonlinearly mixing a smaller number of latent variable distributions. This phenomenon can be explained by the central limited theorem which states that the distribution of a sum of independent variables is prone to a Gaussian distribution. As a result, the latent variables for this numerical process may contain both Gaussian and Non-Gaussian features.

The numerical process is monitored for a period of 7200 sample time. To simulate a fault condition, a step change of magnitude 4 is introduced to the input variable u at sample time 3000. One-thousand normal data samples are first generated to train the PCA, KPCA, KICA, SPA and the proposed three-layer NLGBN model. For PCA, KPCA, KICA, SPA and MWKPCA the number of retained latent variables is set to 3 while for the proposed technique the same number of latent variables is used with two in the middle layer and one in the top layer. In addition, the following radial basis Kernel is used for KPCA, KICA and MWKPCA.

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{c}} \tag{4.26}$$

The Kernel parameter c is set as $c = 500d$ [18,20], where d is the number of monitored process variables. Since only 1000 data samples are provided, the training for the NLGBN has to converge at the 1000th step before conducting the inference step. A series of training rate is tested from 0.002 to 0.02 at 0.002 increments. The training rate achieving the best convergence results for the first case study is 0.01. The convergence plot for the proposed technique with a training rate of 0.01 is presented in Figure 4-5.

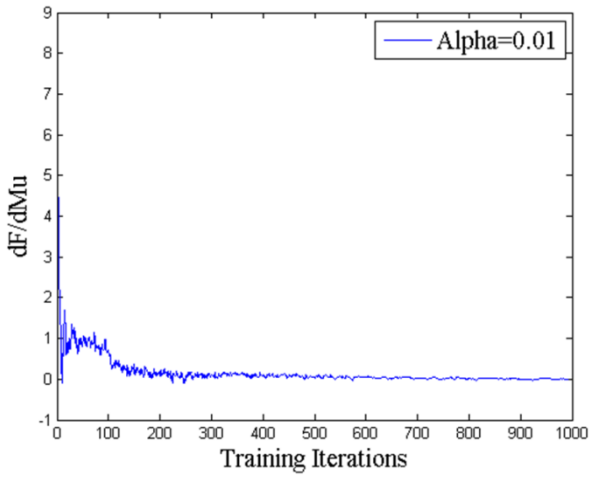

Figure 4-5: Convergence study of the proposed technique for the numerical example

The window size for both the SPA and MWKPCA is chosen to be 100. In total, 10 SP samples are obtained from the 1000 training data samples for the SPA technique. For

each SP sample, 20 statistics are selected according to the criteria suggested in the work of Wang, He[76]. It is noticed that the number of statistics (statistical variables) is twice as much as the number of training SP samples. On the other hand, the KPCA models are obtained online for each monitoring window during process monitoring using the MWKPCA technique. The monitoring window is moving forward in a way such that for each new data sample available, an old data sample is discarded. The monitoring statistics (T^2 and SPE) of the new data sample is computed with respect to the KPCA model built 50 sample intervals earlier. As compared to the other four techniques, MWKPCA is more computationally expensive as the kernel method has to be performed online. The process monitoring charts for these five techniques are shown in Figure 4-6. In addition, the fault detection rates and false alarm rates for these techniques are also summarized in Table 4-2. For PCA, KPCA, KICA, SPA and MWKPCA, both T^2 and SPE statistics are used for fault detection. It may be readily seen that the fault detection performances of KPCA, KICA, SPA and the NLGBN-based technique are very close to each other (with almost 100% fault detection efficiency) which indirectly proves that the latent variables of the process may contain both Gaussian and non-Gaussian features. On the other hand, PCA produces less accurate results (only around 73% based on T^2 statistic) as it neither can capture and nonlinear relationship nor can deal with noise in data. The MWKPCA technique performs the worst with approximately only 20% of detection rate. This is mainly due to the fact that the system is stabilized after the fault, as shown in Figure 4-7. The KPCA models built for the faulty data samples do not vary significantly from each other. As a result, a new faulty data samples might be classified as normal with respect to the KPCA model built 50 steps earlier. In contrast, NLGBN-based technique is able to incorporate the effect of noise allowing it to effectively extract useful features from noisy data; therefore, the process monitoring charts are much smoother. This leads to similar fault detection performance of the proposed technique as compared to the KPCA and KICA but at a lower computational cost. For the same reason, the proposed technique also has the lowest false alarm rate.

Table 4-2: Fault detection rates and false alarm rates for the numerical process

Fault Detection Rate (%) under 95% Confidence UCL										
PCA		**KPCA**		**KICA**		**SPA**		**MWKPCA**		**NLGNB**
T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	***G-Index***
72.88	5.02	7.55	99.95	99.89	99.74	98.86	99.81	20.95	23.00	99.65
False Alarm Rate (%) under 95% Confidence UCL										
PCA		**KPCA**		**KICA**		**SPA**		**MWKPCA**		**NLGNB**
T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	***G-Index***
0.97	1.31	2.34	1.23	1.20	5.23	4.78	4.52	3.78	4	**0.80**

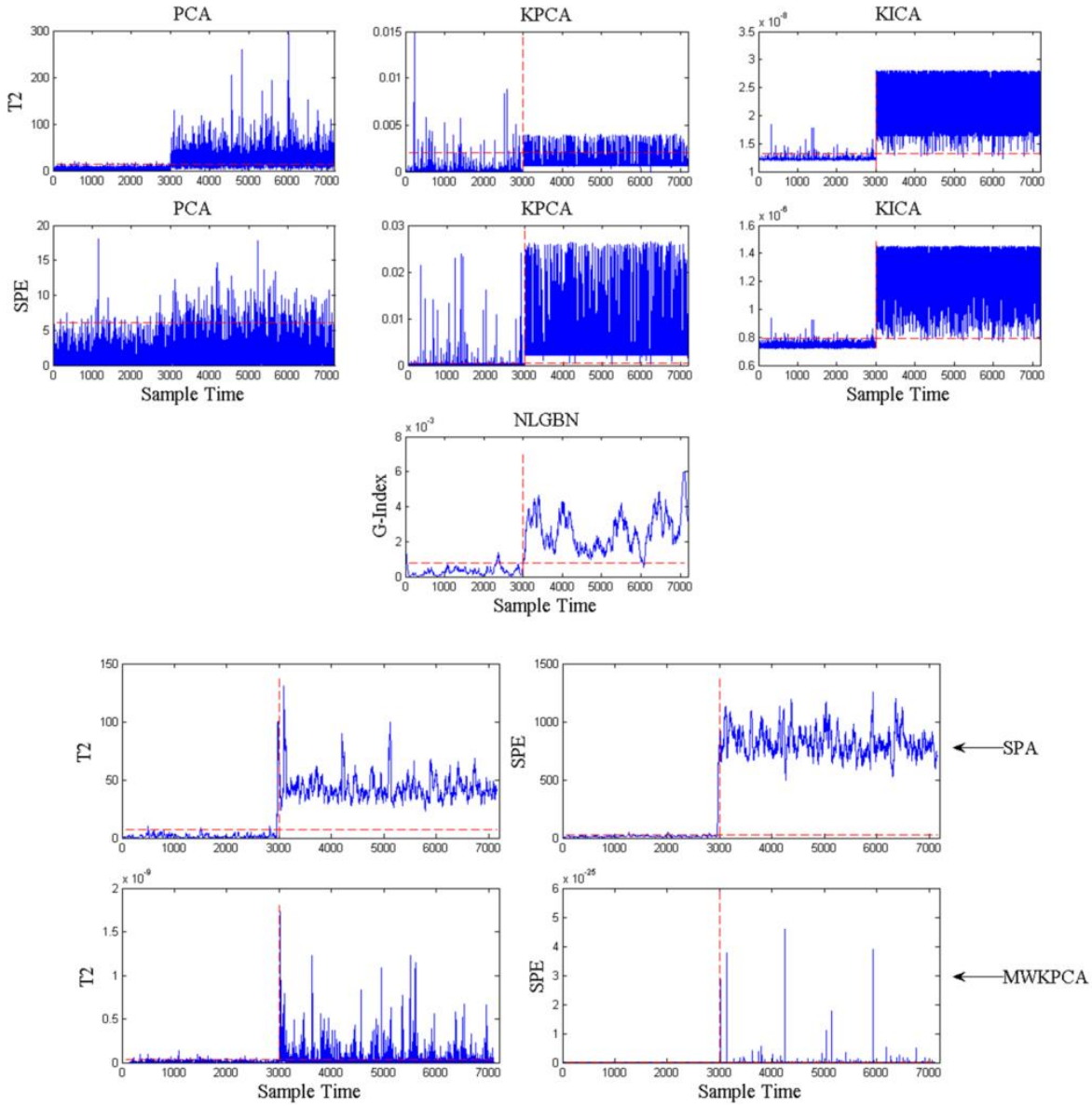

Figure 4-6: Process monitoring charts of PCA, KPCA, KICA, SPA, MWKPCA and the NLGBN-based technique

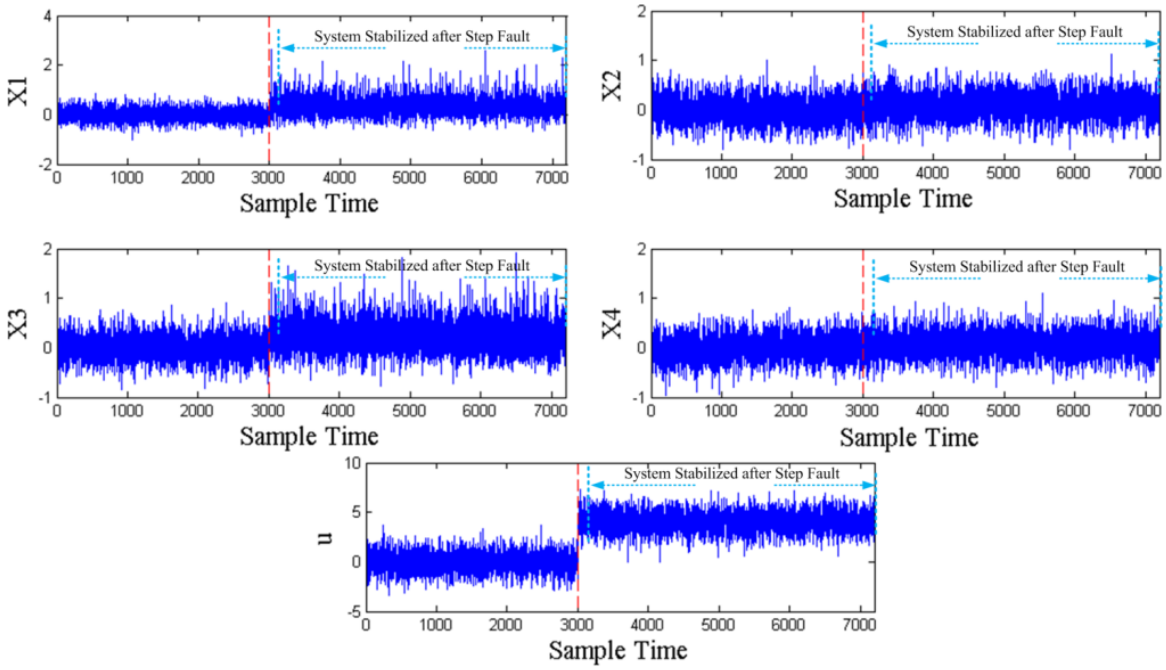

Figure 4-7: Dynamical behaviour of process variables after STEP fault

For fault diagnosis, the comparison is only made between PCA and the proposed technique as the fault diagnosis techniques based on KICA, KPCA, MWKPCA and SPA have yet to be developed. The contribution plots are generated based on the average contribution of each process variable across the first 100 data samples after sample interval 3000 when the fault is injected into the system. As shown in Figure 4-8, both PCA and the proposed technique are able to correctly identify the root-cause variable under the step-change fault. However, for PCA, the other irrelevant process variables also have high contribution to the fault giving non-robust diagnosis. Additionally, the poor performance of PCA in fault detection further limits its capability. As demonstrated in this case study, the proposed technique outperforms the conventional techniques in both fault detection and diagnosis.

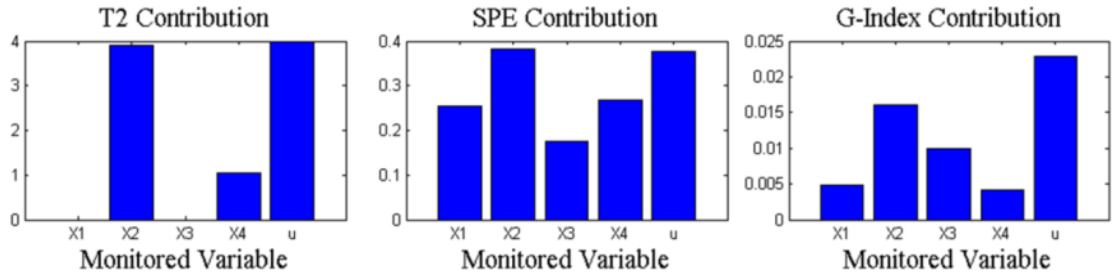

Figure 4-8: Comparison of fault diagnosis performance between PCA and the NLGBN-based technique

4.4.2 Tennessee Eastman chemical process

In this section, the effectiveness of the proposed NLGBN-based online fault diagnosis is verified using on a well-established Simulink simulation package of the Tennessee Eastman chemical process. The simulation adopts the decentralized control strategy to

construct a closed-loop stable simulation of the process.[86] The Tennessee Eastman chemical process comprises of five major operating units: an exothermic two-phase reactor, a product condenser, a vapour-liquid flash separator, a recycle compressor, and a reboiled product stripper.[32] The process flow diagram of the chemical plant is shown in Figure 9-1. In total, there are 41 measured process variables in the process, among which 22 variables are monitored to determine the operating condition of the process system. These monitored variables are listed in **Error! Reference source not found.**.

In addition, Table 9-2 summarizes the 15 fault conditions that have been pre-programmed in the Tennessee Eastman process simulation and have been widely used by the process monitoring community for verifying and comparing various techniques [32]. In this study, 4 of these conditions (IDV6, IDV7, IDV10 and IDV11) are used to verify the proposed fault diagnosis technique. The sampling interval for data collection is 0.05 hr.

Similar to the first case study, one-thousand normal process data samples are used to train the five fault diagnosis techniques: PCA, KPCA, KICA, SPA and NLGBN. In case of KICA and KPCA, the eigenvectors having a Eigen value satisfying the following condition is chosen for feature extraction.[20]

$$\frac{\lambda_i}{\sum_i \lambda_i} > 0.0001 \tag{4.27}$$

Where λ_i is the Eigen value corresponding to the i^{th} eigenvectors. As a result, 9 eigenvectors are persevered for both KICA[‡‡] and KPCA. Likewise, the Kernel parameter for the radial basis Kernel used is set to $c = 500d$ [18]; d is the number of monitored variables which is equal to 22 in this case study. To establish a consistent basis for comparison, the same number of latent variables is used for PCA, SPA, MWKPCA and the three-layer NLGBN with 8 latent units in the middle layer and 1 in the top layer. The convergence plot with a training rate of 0.008 for the variational training of the three-layer NLGBN is shown in Figure 4-9. This training rate is determined in the same way as in the first case study. In this case study, to generate more training sample for the SPA and MWKPCA technique, the window size is reduced to 50 samples; the number of SP training samples is 20. In addition, 162 statistics are selected for each SP sample, which is significantly larger than the number of training samples.

[‡‡] KICA uses KPCA for Kernel whitening and also for determination of the number of eigenvectors. This is why KPCA and KICA have the same number of eigenvectors.

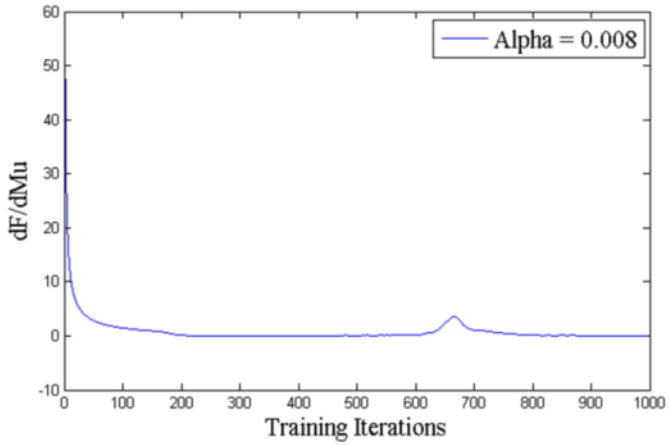

Figure 4-9: Convergence study of the proposed technique for the Tennessee Eastman process

The fault detection performance in terms of fault detection rates and mean false alarm rates[§§] for these four techniques are summarized in Table 4-3.

[§§] Determined by averaging the false alarm rates of all 15 cases.

Table 4-3: Fault detection rates and mean false alarm rates for the TE process

Fault Detection Rate (%) under 95% Confidence UCL											
Faults	**PCA**		**KPCA**		**KICA**		**SPA**		**MWKPCA**		**NLGBN**
	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	***G-Index***
1	99.76	99.86	100	100	100	100	98.86	99.81	36.24	97.33	100
2	99.43	99.60	99.19	99.21	99.02	100	78.83	98.86	6.95	96.88	100
3	2.09	1.56	2.76	0.024	8.83	1.40	0.1	1.74	2.14	0.26	***31.25***
4	0.45	1.07	1.41	0.31	1.43	1.33	0.33	2.36	3.64	0.74	***45.49***
5	1.43	1.19	0.9	0	1.38	1.83	0.24	1.48	1.55	0.21	***14.28***
6	99.86	100	43.24	100	97	100	95.37	100	31.14	37.07	100
7	2.45	4.18	5.76	4.85	2.33	5.02	2.9	15.88	2.75	1.45	**32.35**
8	91.32	94.36	91.69	97.98	75.37	92.89	98	97.76	96.05	95.64	98.31
9	1.09	0.93	2.38	0	1.48	1.74	2.38	10.62	10.50	0.74	***30.35***
10	1.59	40.62	24.16	74.63	67.76	70.88	58.38	51.43	74.62	68.38	***80.58***
11	27.80	51.24	***84.13***	54.55	54.17	80	79.90	39.02	77.86	81.48	79.39
12	34.85	21.35	35.09	25.07	37.95	9.81	13.4	41.6	***66.81***	25.55	64.77
13	84.10	93.03	91.67	93.17	91.03	96.13	92.81	95.12	91.86	90.86	94.56
14	25.56	99.67	98.52	95.17	97.40	100	98.86	34.76	97.29	96.86	100
15	1.12	1.21	1.07	0	2.95	3.67	0.1	1.74	4.19	2.98	***10.00***
Mean False Alarm Rate (%) under 95% Confidence UCL											
	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	T^2	***SPE***	***G-Index***
	1.37	0.7	1.20	1.28	2.27	1.27	1.28	2.42	2.21	1.52	***0.61***

It may be seen that the performance of KPCA and KICA is quite close to each other considering both T^2 and SPE statistics across all 15 cases. In fact, the similar results have also been reported in the work of [18]. KICA and KPCA represent two extreme cases each of which assumes the latent variables have either pure non-Gaussian distribution or pure Gaussian distribution. In the first case, KICA should outperform KPCA on a consistent basis. Similarly, in the second case, KPCA should take dominance. However, the results of this case study indicate that the latent variables may have a combination of both Gaussian and non-Gaussian features enabling both KPCA and KICA to perform at approximately the same level of accuracy. The SPA and MWKPCA techniques take autocorrelation and cross-correlation of the data samples into consideration. In comparison to PCA, KPCA and KICA, the fault detection rates of these two techniques are higher on average. The SPA method performs better for step fault conditions while the MWKPCA achieves higher fault detection rate for faults with increased random variation. This observation is in consistence with the first case study. The step fault conditions are static as compared to the random variation fault conditions. The closed-loop controller of the TEP process can quickly adapt to these fault conditions and stabilize the system. Once the system is stabilized, the MWKPCA generates KPCA models with high similarity for each online monitoring window leading to low fault detection rate. On the other hand, the T^2 and SPE statistics of the SPA technique are determined with respect to the models obtained offline from historical data samples; they are still able to identify the breakdown of correlation structure even when the system is stabilized upon fault. The SPA technique performs much worse than the MWKPCA for faults with randomly changing magnitude, in particular for IDV10, 11 and 12. This is due to the fact that the number of statistical variables is much larger than the number of SP training samples. The scarce of training samples leads to non-robust feature extraction.

In comparison, the proposed technique which takes into account the noisy nature of the data and utilizes the G-Index which is sensitive to subtle disruption of problem space due to fault has seen significant improvement in performance in the fault conditions 3, 4, 5, 7, 9, 10, and 12. Particularly, for fault conditions 3 and 9 which have been reported by many studies [15,76,87,88] to be difficult to detect, the proposed technique is able to identify more than 30% of fault samples for both fault conditions. Additionally, the proposed technique also has the lowest mean false alarm rate at only 0.61%. In a clear contrast, PCA, KPCA and KICA are not able to detect these trivial faults in this case study (only single digit fault detection rates are observed for both T^2 and SPE statistics). It is noticed that PCA has performed poorly on majority of the conditions due to the fact that it cannot deal with noisy data and process nonlinearity. To better demonstrate the performance of the proposed technique to KPCA, KICA, SPA and MWKPCA in fault detection, the monitoring charts of fault condition 10 and 12 for all these techniques are shown in Figure 4-10 and Figure 4-11, respectively.

It may also be observed the process monitoring charts generated by NLGBN are much smoother than those of PCA, KPCA and KICA. This is due to the fact that NLGBN effectively incorporates the effect of noise in data which leads to a robust feature extraction.

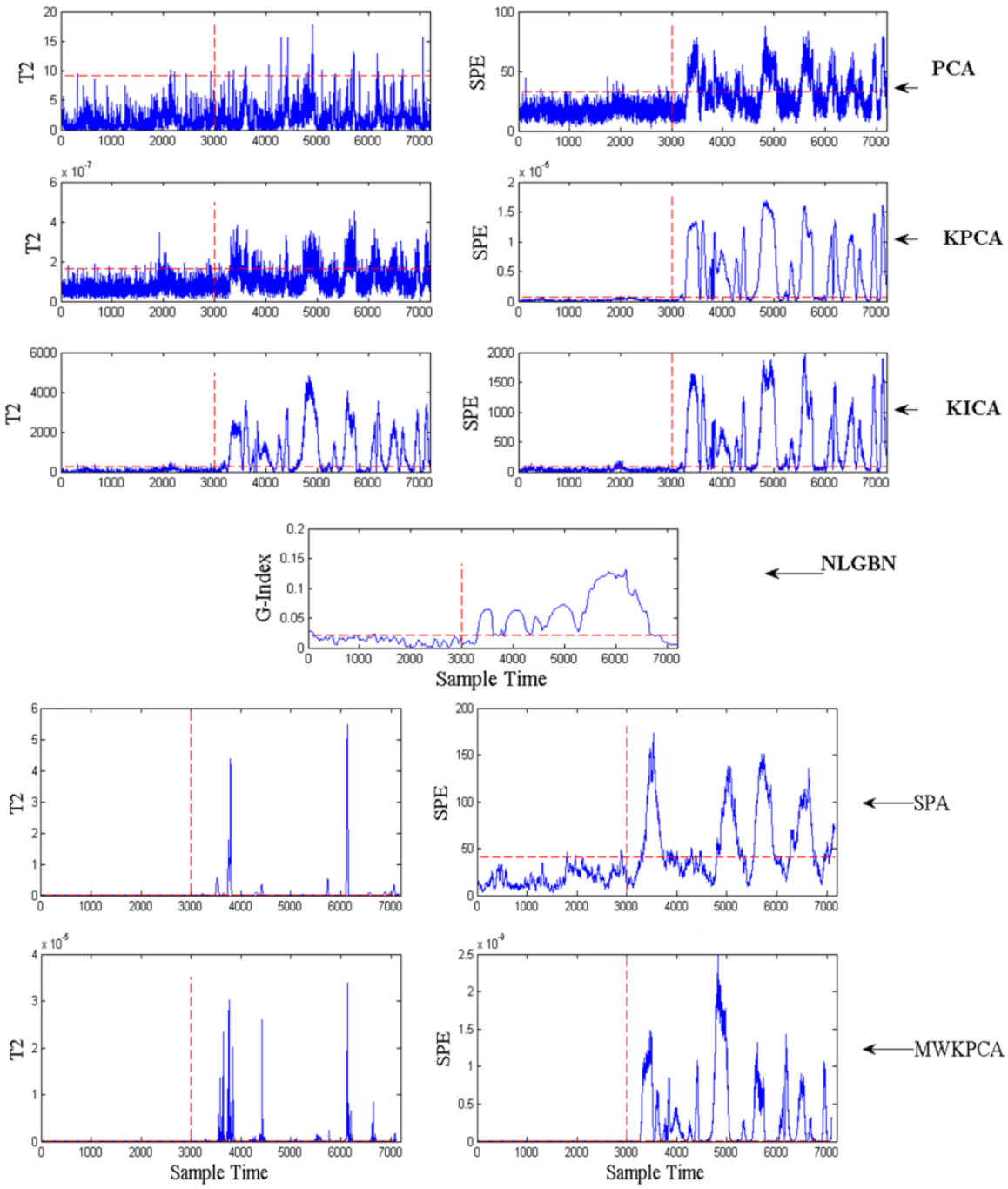

Figure 4-10: Process monitoring charts for IDV10 based on PCA, KPCA, KICA, SPA, MWKPCA and NLGBN

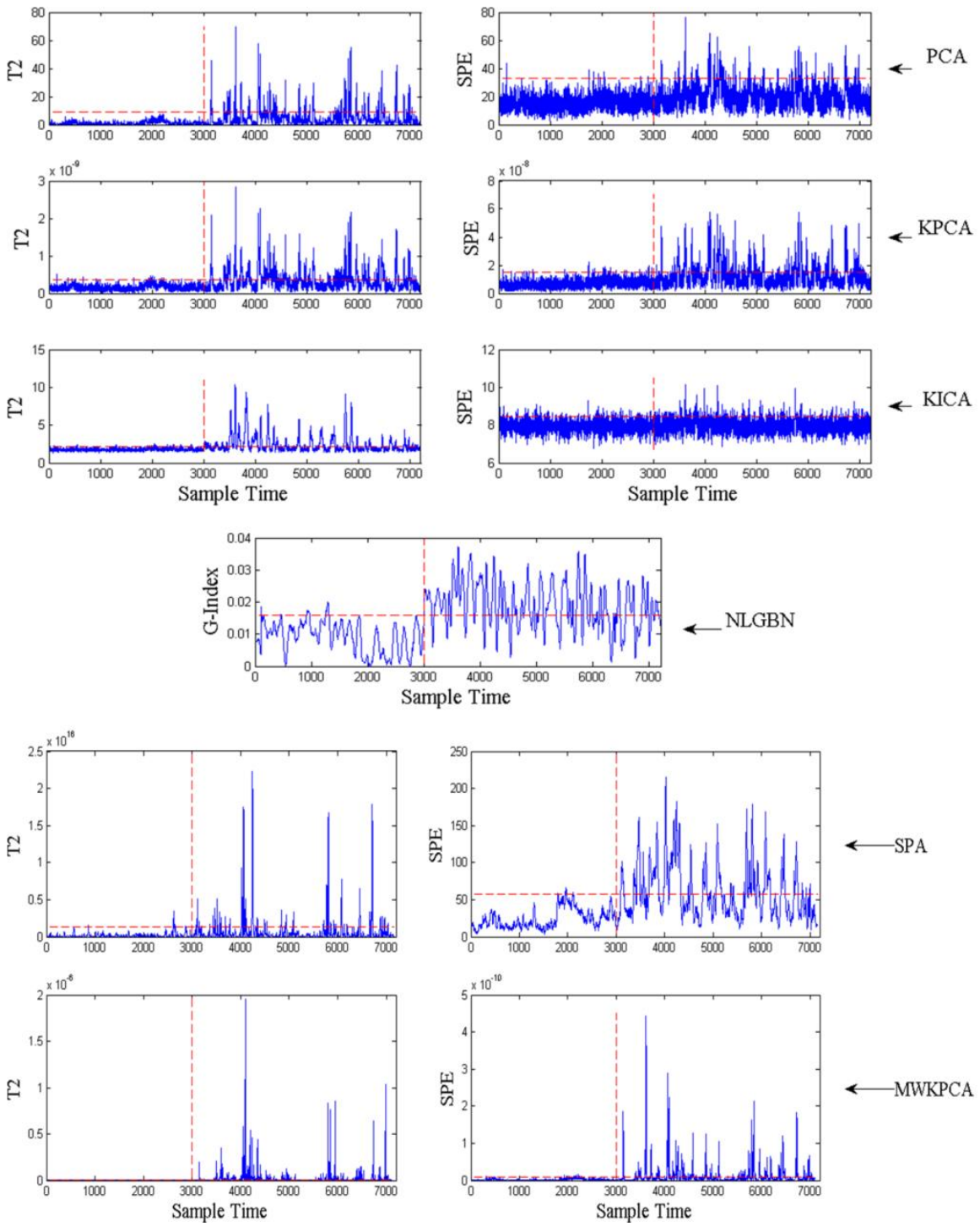

Figure 4-11: Process monitoring charts for IDV12 based on PCA, KPCA, KICA, SPA, MWKPCA and NLGBN

For fault diagnosis, the performance of the NLGBN-based technique is only compared to that of the PCA as it is practically impossible to determine individual contribution of process variable through reverse projection with KPCA and KICA. The fault diagnosis results for IDV6, IDV7, IDV10 and IDV11 are presented in Figure 4-12. In fault condition IDV6, a step change is introduced to the feed A (X1) of the process. The SPE statistic contribution and G-Index contribution correctly identify the root cause variables;

feed A has the highest contribution to the fault. On the other hand, the T^2 statistic fails to identify the closely related root-cause variable. In the second fault condition IDV7, a step change has been introduced to the C header pressure which in turn reduces the availability of feed C (Stream 4). Due to the pressure loss in Stream 4, the stripper pressure (X16) is the only monitored process variable that is directly related to this fault condition. As shown in Figure 4-12, the abnormal behaviour of the stripper is correctly identified as it has the highest contribution to the fault. Moreover, as the stripper is a crucial operating unit in the recycle operation, this fault effect is also propagated to the stripper temperature (X18) and to upset the reactor pressure (X7) and separator (X13) through the recycle loop shown in **Error! Reference source not found.**. In comparison, the PCA-based echniques fail in identifying the root-cause variable. Similarly, in fault condition 10, due to the random variation of temperature in feed C, the temperature of the stripper column is directly affected (X18). This abnormal variation in stripper temperature is correctly captured by the proposed technique. However, this random variation also introduces additional noise in data and this noise is further disrupted by the nonlinear interaction between the process variables leading to poor performance of the PCA-based techniques. In the last case, the random variation in reactor cooling water inlet temperature has resulted in abnormal behaviour of the reactor temperature (X9). Since the cooling water inlet temperature is not a monitored process variable, the proposed technique as a multivariate data analysis technique is not able to identify this true root-cause process variable. Nevertheless, the most closely related monitored process variable, reactor temperature (X9), is successfully located. In contrast, the PCA-based techniques are incapable of locating this variable.

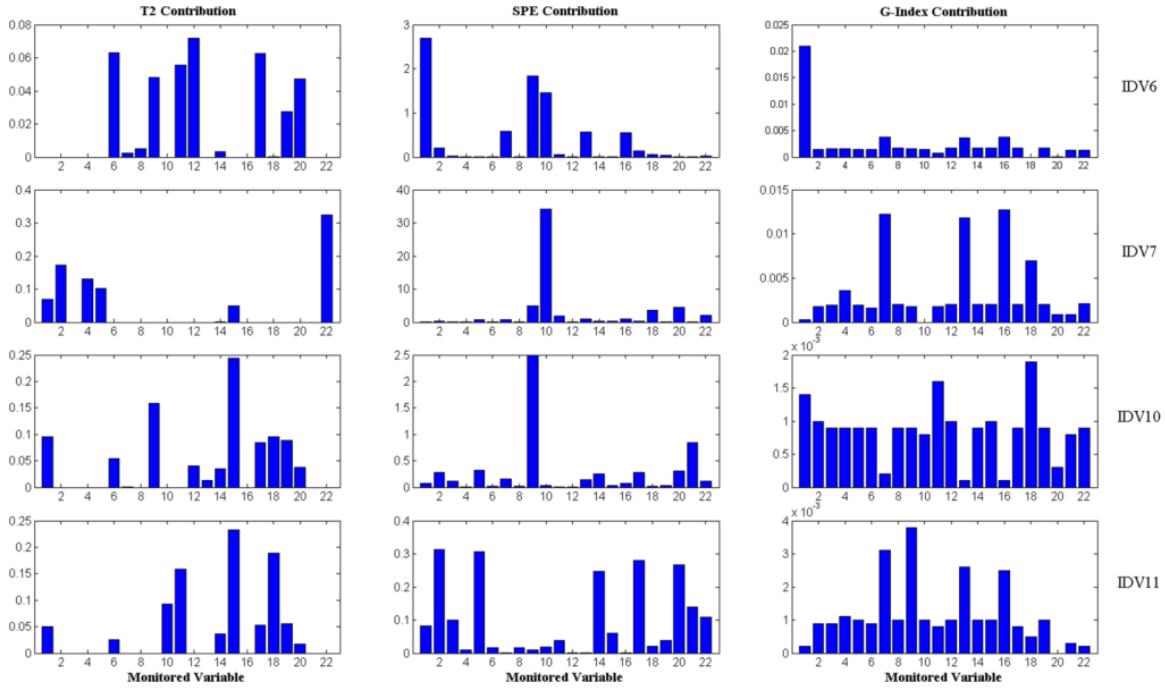

Figure 4-12: Fault diagnosis results of PCA-based techniques and the proposed NLGBN-based technique

4.5 Conclusion

In this study, a NLGBN-based online fault diagnosis technique is proposed for industrial processes. A three-layer NLGBN is constructed and trained to extract useful features from noisy process data. The nonlinear relationships between the process variables and latent variables are captured by using a set of sigmoidal functions between the middle layer and the bottom layer. The middle layer units are then connected to a single top-layer unit through a set of linear weights. A set of model variances is also introduced to each layer to take into account the noisy nature of the data. The network is trained until the gradient to update the posterior mean of the top-layer unit converges at a value near zero. During online monitoring, the model parameters of the NLGBN are fixed and the online process data is used to update the posterior mean of the top-layer latent unit again. The absolute value of the gradient updating the posterior mean is monitored for fault detection. The main advantage of monitoring the gradient is that it is a measure of the local curvature of the highly nonlinear problem space which makes it sensitive to the subtle disruption introduced by any fault condition. For fault diagnosis, a multivariate contribution plot is generated based on the contribution of each process variable to the G-Index. The effectiveness of the proposed technique is demonstrated on two case studies. It is shown that when the latent variables of industrial processes have both Gaussian and non-Gaussian features, the ability to extract only non-Gaussian features do not yield significant improvement in performance of fault diagnosis technique. On the other hand, a significant improvement in fault detection accuracy is observed for the proposed technique which takes into account the noise nature of the process data. The NLGBN-based technique also addresses the issue of intractable reverse projection of the KPCA,

KICA and MWKPCA. This leads to robust fault diagnosis on nonlinear and noisy industrial processes.

5 Modified Independent Component Analysis and Bayesian Network based Two-stage Fault Diagnosis of process operations

Abstract

Statistical fault detection techniques are able to detect fault and diagnose root-cause(s) from the monitored process variables. For complex process operations, it is not feasible to screen all the process variables due to monitoring cost and flooding of alarms. Thus if a fault is originated from a process variable that is not monitored, conventional statistical techniques are incapable of locating the true root-cause. To relax this limitation, a two-stage fault diagnosis technique is proposed for process operations. In the first-stage, the modified independent component analysis is used for fault detection and to identify the faulty monitored variable. In the second-stage, a Bayesian Network model is constructed considering the process variables and their dependence obtained from the process flow diagram. Evidence is then generated at the network node corresponding to the faulty variable identified in the first-stage. Subsequently, the network is updated and analysed using deductive and abductive reasoning to identify the true root-cause. To verify the applicability of the proposed technique it is tested on two process models. The results of both case studies have demonstrated the effectiveness of the proposed technique to diagnose the true root-cause originated from process variables that are not monitored. Once integrated with process loss functions, the proposed technique will serve as an important element of dynamic operational risk management framework.

Keywords: Process operations, fault diagnosis, modified independent component analysis, Bayesian network

5.1 Introduction

Modern industrial processes are large-scale systems that comprise of many operating units and multiple processing steps to produce high quality products. To ensure the safety of production and the personnel involved, industrial processes are monitored on a real-time basis. This requires the online measurement of a large number of process variables associated with various process components. Due to the complex nature of process operation, functions that govern the relationship amongst the process variables are often high-order and nonlinear, and are difficult to obtain explicitly. As a result, the conventional first-principle-model-based process monitoring techniques become less suitable.[65] To relax this limitation, multivariate statistical process monitoring (MSPM) techniques have been proposed to extract features or latent variables from the highly-correlated and high-dimensional process data to detect and diagnose various faults of industrial processes.[2-4,7,33,66]

Principle Component Analysis (PCA) and Partial Least Square (PLS) are the most extensively used statistical feature extraction techniques for process monitoring.[7-10,67] These techniques implicitly assume that process variation follow a multivariate Gaussian distribution and determine a set of orthogonal projection vectors called loading vectors. Through these loading vectors, process data can be projected into a subspace or feature space with lower dimensionality.[89] These loading vectors represent the directions of most significant variability of the process data.[6,90] In addition, due to the orthogonal transformation, the cross-correlation (linear-correlation) between the process variables are removed so that a new set of pairwise independent variables known as the principal components (PCs) are formed into the feature space.[69,91] In this regard, PCA and PLS only manipulate the second-order statistics (correlation and cross-correlation) of the process data.[92,93] For process monitoring, the Hotelling's T^2 statistics and squared prediction error (SPE) statistics of the PCs are computed to detect process abnormalities. The T^2 and the SPE statistics have different physical meaning and cover different aspects of fault detection. T^2 measures the correlated distance from the centre of the feature space to the projected data sample. On the other hand, the SPE statistics is a L2 norm which measures the Euclidean distance from the residual space to the PC feature space. In other words, the T^2 statistics measure the systematic variation among data while the SPE statistics measure the residual variation. The control limits for both statistics are derived based on the assumption that the PCs follow Gaussian distribution, in particular for SPE, a standard normal distribution.[72] Multivariate contribution plots are also generated based on these two statistics to isolate the root-cause variable. However, in complex industrial processes, the behaviour of a process variable can be affected by a number of other process variables. The second-order statistics that describes only pairwise relationship can become inadequate for feature extraction. Furthermore, the process data do not always follow Gaussian distribution due to process non-linearity and external disturbance. As a result, PCA/PLS based techniques may produce misleading results for performance monitoring of complex industrial processes.[76]

Independent Component Analysis (ICA) has been proposed to address the limitation of PCA/PLS based techniques.[1] ICA determines a set of non-orthogonal demixing vectors

through which the process data can be transformed into a subset of independent components (ICs) that have minimum mutual information.[16,94] Mutual information is a measure of difference between the joint distribution and the marginal distributions of the ICs.[95] Thus, mutual information takes into account the complete dependence structure of the latent variables rather than second-order dependence of the PCA.[16] Moreover, mutual information is equivalent to the well-known Kullback-Leibler divergence that measures the difference in entropy between the joint distribution and marginal distributions of the latent variables.[96] To minimize the entropy difference, the latent variables have to be not only as independent as possible but also as non-Gaussian as possible.[16] In this regard, ICA explores high-order statistics and retains non-Gaussian features of the process; thus, it yields better results as compared to PCA for complex process monitoring.[1,92] However, one of the major drawbacks of the conventional ICA is that the extracted ICs are of the same importance. It is therefore difficult to determine the dominant ICs for dimensionality reduction. The modified ICA is then developed to solve this problem by preserving the ranking of PCs in the PCA whitening step.[15] Subsequently, the ICA version of T^2, also known as the I^2 statistics, and the SPE statistics similar to PCA are computed for process monitoring. Since the ICs do not follow Gaussian distribution, the kernel density estimation is adopted to estimate the control limits for these two statistics.[1] Similar to PCA, multivariate contribution plots can be generated based on the I^2 and SPE statistics to locate the root-cause process variable.

PCA, PLS and ICA do not require any prior knowledge of the process but rely heavily on availability of online monitored data. These techniques are inadequate to isolate root-cause from process variables that are not monitored. In industrial practice, abundant number of monitored variables may substantially increase rate of false alarms. In addition, some of the process variables are very costly to monitor as they may be associated with operating units that are located in congested space and require very sophisticated measuring instruments. Furthermore, the malfunction of these measuring instruments can produce misleading results that disguise the true root-cause of the process abnormality. To overcome the above problems, the number of monitored process variables is restricted, and therefore cannot cover full aspects of the process operation. In this case, statistical data-driven techniques may not be able to point to the true root-cause of the process abnormality. Bayesian Network (BN) can be used as an efficient tool to allay the limitation of statistical data-driven techniques. The BN utilizes the prior process knowledge to construct directed-acyclic-graphic representation of the process.[97] Unlike the conventional model-based techniques, BN require only the causal relationship among the process variables which is relatively easy to obtain by analysing the process flow diagram. Each process variable is represented as a random variable node in the graphic structure. These nodes are connected by arcs that describe the casual dependence amongst the process variables. In addition, the statistical dependence among the process variables is quantified by the probabilistic measure of influence of one process variable on another. Two types of logical reasoning are incorporated with BN, namely deductive reasoning and abductive reasoning.[98] Deductive reasoning allows inference of the states of the unobserved process variables given the state of an observed process variable. On the other hand, abductive reasoning, also known as the most probable explanation (MPE),[99]

determines the most likely combination of states of various unobserved process variables that best explain the state of the observed process variable. Utilizing these reasoning mechanisms, BN is able to isolate the true root-cause from unobserved (not monitored) process variables.

In this study, a combination of modified ICA and BN is applied for two-stage fault diagnosis technique of industrial processes. Here, the modified ICA identifies the faulty monitored variable and is used as the evidence node in BN for finding the true root-cause amongst the intermediate variables through deductive and abductive reasoning approaches. The proposed two-stage fault diagnosis technique is demonstrated with two illustrative examples.

5.2 Background

5.2.1 Independent Component Analysis

ICA is able to extract statistically independent components from highly-correlated and high-dimensional data and has been widely applied for blind source separation and signal separation.[100] In recent years, ICA has also been extensively applied to monitor complex industrial processes. ICA performs better than many conventional techniques due to its ability to extract non-Gaussian features which are considered to be dominating in modern processes.[101-103] For process monitoring, the monitored process data can be considered as a linear combination of signals that are generated from a subset of independent sources or latent variables. Suppose a process data matrix $\mathbf{X} \in \mathbb{R}^{d \times n}$ having *d* process variables and *n* samples is generated from the normal operating condition of a process. The ICA decomposition of X is expressed as

$$\mathbf{X} = \mathbf{AS} + \mathbf{E} \tag{5.1}$$

where **A** and **E** are the mixing matrix and the residual matrix respectively. $\mathbf{S} = [s_1^n, s_2^n, s_3^n, ..., s_p^n] \in \mathbb{R}^{p \times n}$ is the independent component matrix consisting of p ($p<d$) independent components. Both **A** and **S** are estimated from **X** through an iterative algorithm, known as the fast fixed-point ICA algorithm (FastICA) [73]. Data matrix **X** is first whitened through PCA whitening which removes the pairwise dependence among process variables.

$$\mathbf{Z} = \mathbf{D}^{-1/2}\mathbf{V}^{\mathbf{T}}\mathbf{X} \tag{5.2}$$

where **D** and **V** are the eigenvalue matrix and eigenvector matrix of the covariance matrix of **X** respectively, $E\left[\mathbf{X}\mathbf{X}^{\mathbf{T}}\right] = \mathbf{V}\mathbf{D}\mathbf{V}^{\mathbf{T}}$. $\mathbf{Z} = [z_1^n, z_2^n, z_3^n, ..., z_p^n] \in \mathbb{R}^{p \times n}$ is the whitened data matrix in the ICA subspace. It is noted that $E\left[\mathbf{Z}\mathbf{Z}^{\mathbf{T}}\right] = \mathbf{D}^{-1/2}\mathbf{V}^{\mathbf{T}}E\left[\mathbf{X}\mathbf{X}^{\mathbf{T}}\right]\mathbf{V}\mathbf{D}^{-1/2} = \mathbf{I}$.

The whitened data is then projected into the ICA feature space through the following transformation.

$$\hat{\mathbf{S}} = \mathbf{WZ} \tag{5.3}$$

where $\mathbf{W} = \mathbf{A}^{-1}$ is the normalized demixing matrix and $\hat{\mathbf{S}} = [\hat{s}_1^n, \hat{s}_2^n, \hat{s}_3^n, ..., \hat{s}_p^n] \in \mathbb{R}^{p \times n}$ is the estimated independent component matrix. The objective of the FastICA algorithm is to find the $\mathbf{W}$ that maximizes the non-Gaussianity of each element $\hat{s}_p^n$ of $\hat{\mathbf{S}}$ such that they become as independent as possible. Considering a single independent component $\hat{s}_i^n, i \in \{1, 2, 3, ..., p\}$.

$$\hat{\mathbf{s}}_i^n = \mathbf{w}_i^{\mathrm{T}} \mathbf{z}_i^n \tag{5.4}$$

where $\mathbf{w}_i$ is the i^{th} column vector of $\mathbf{W}$. The non-Gaussianity of $\hat{\mathbf{s}}_i^n$ is measured by its negative entropy, *Negentropy*, which is approximated as follows:

$$J(\hat{\mathbf{s}}_i^n) \propto \left[E\{G(\hat{\mathbf{s}}_i^n)\} - E\{G(\mathrm{v})\} \right]^2 \tag{5.5}$$

where v is the standardized Gaussian variable and G is a non-quadratic function that normally takes the following two forms [73].

$$G_1(u) = \frac{1}{a_1} \log \cosh a_1 u, \quad G_2(u) = -\exp(-u^2/2) \tag{5.6}$$

Subsequently, $\mathbf{w}_i$ is obtained by iteratively maximizing the following objective function.

$$\mathbf{w}_i = \arg\max_{\mathbf{w}_i} \left[E\{G(\mathbf{w}_i^{\mathrm{T}} \mathbf{z}_i^n)\} - E\{G(\mathrm{v})\} \right]^2 \tag{5.7}$$

5.2.2 Bayesian Network

Bayesian network is a powerful tool for modelling complex systems owing to its flexible structure and robust reasoning capability [104,105]. Bayesian Network (BN) is a graphic model that consists of a set of nodes being connected with directed arcs. Each node represents a random variable and the arcs indicate the causal relationships among the random variables. The direction of the arcs determines the dependence of one variable on another. For a pair of nodes, the node from which the arc is directed is the ancestor node; while the node receiving the arc is the descendent node. An arc from a descendent node can never return to any of its ancestor nodes. The nodes that do not have any descendent nodes are referred as leaf nodes. In contrast, the nodes without any ancestor nodes are referred as the root nodes. BN also satisfies the local Markov property which dictates the dependence structure among nodes; a node is conditionally independent of any of its non-descendent nodes given the state of its direct ancestor node [106]. This property of BN allows factorization and efficient computation of the joint probability distribution of the random variables within the graphic model.

In addition to the qualitative causal reasoning provided by the graphic model, a set of the parameters which describe the statistical dependence amongst the random variables,

denoted $\theta_i \in \Theta$, are to be estimated. Considering the following BN model in Figure 5-1(a) which has 5 nodes and 4 arcs. An alternative representation of this BN is a factorized model shown in Figure 5-1(b). The arcs that connect the ancestor nodes and the descendant nodes are joined by a factorial node; e.g. X_1, X_2 and X_3 are joined by f3.

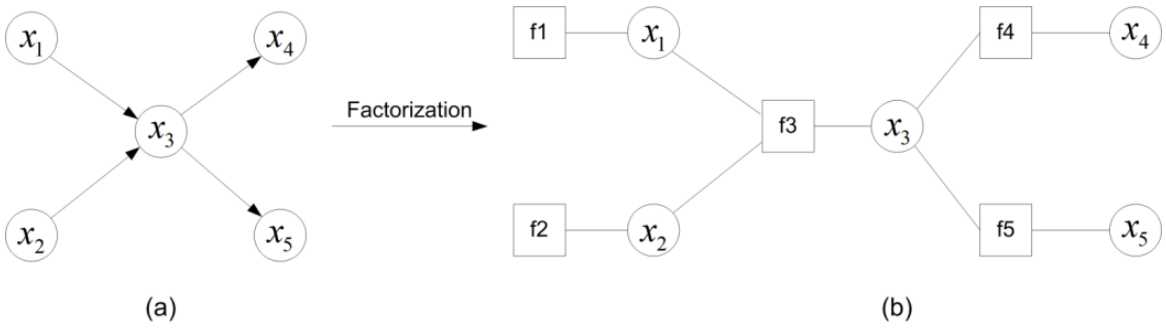

Figure 5-1: Bayesian Network Factorization.

Each factorial node represents the statistical dependence between the ancestor nodes and descendent nodes.

$$\begin{aligned} &f_1(x_1) = P(x_1), \quad f_2(x_2) = P(x_2); \\ &f_3(x_3, x_1, x_2) = P(x_3 \mid x_1, x_2); \\ &f_4(x_4, x_3) = P(x_4 \mid x_3), \quad f_5(x_5, x_3) = P(x_5 \mid x_3). \end{aligned} \tag{5.8}$$

The joint probability distribution of the BN is therefore expressed in terms of product of factorials.

$$\begin{aligned} &P(x_1, x_2, x_3, x_4, x_5) \\ &= f_1(x_1) f_2(x_2) f_3(x_3, x_1, x_2) f_3(x_3, x_1, x_2) f_4(x_4, x_3) f_5(x_5, x_3) \end{aligned} \tag{5.9}$$

In more general form, the joint probability distribution for a BN with ***K*** factorials is expressed as

$$P(\mathbf{X} \mid \mathbf{\Theta}) = \prod_{k=1}^{K} f_k(x_j, x_i \mid \theta_k) \tag{5.10}$$

Where $\mathbf{X}$ is the set of random variables in the BN. x_j and x_i are the descendent nodes and ancestor nodes in each factorial respectively, $x_i, x_j \in \mathbf{X}$. $\theta_k \in \mathbf{\Theta}$ are the probability density functions (CPDFs) associated with each factorial. The CPFD describes the likelihood of a given state of the ancestor nodes leading to a particular state of the descendent node. In this study, the states of the network nodes are discrete and are classified into normal (State 1) and faulty (State 0) in the case studies section; θ_k can be used to generate a conditional probability table (CPT) for each pair of ancestor and descendent nodes. Given a set of training data $\tilde{\mathbf{X}}$ which covers all possible combinations

of states of the nodes in BN, the likelihood of the training data $\tilde{\mathbf{X}}$ given the BN structure and factorials is expressed as:

$$P(\tilde{\mathbf{X}} \mid \boldsymbol{\Theta}) = \prod_{k=1}^{K} f_k(\tilde{x}_j, \tilde{x}_i \mid \theta_k) \tag{5.11}$$

Where $\tilde{x}_j$ and $\tilde{x}_i$ are the training data for each ancestor node and descendent node, $\tilde{x}_i, \tilde{x}_j \in \tilde{\mathbf{X}}$. The set of factorial CPDFs $\hat{\boldsymbol{\Theta}}$ is estimated using Maximum Likelihood Estimation which determines the $\boldsymbol{\Theta}$ that maximizes the joint log likelihood of the training data. The objective function for this maximization problem is given as:

$$\hat{\boldsymbol{\Theta}} = \arg\max_{\boldsymbol{\Theta}} \sum_{i=1}^{k} \log f_k(\tilde{x}_j, \tilde{x}_i \mid \theta_k) \tag{5.12}$$

The above maximization can be further simplified by considering the Markov properties of the BN. Specifically, the log likelihood for each pair of ancestor nodes and descendent nodes can be maximized independently of other pairs of nodes in the network. As the states are discrete, the CPDFs can be estimated by simply counting the frequency that a given state in the ancestor node links to the states in the descendent node. The detailed BN model training steps can be found in the book of Nielsen, Jensen[107] and Murphy[108]. After the determination of the network model and the associated parameters, Sum-product algorithm is used for deductive reasoning to infer the states of unobserved nodes given the states of observed nodes. Then, max-product algorithm is used for abductive reasoning to determine the most likely combination of states of the unobserved nodes that best explain the observations.

5.3 Methodology

5.3.1 Modified Independent Component Analysis for First-stage Fault Diagnosis

The process variables are grouped into input variables, intermediate variables and output variables. The states of the input variables are relatively independent of the other variables in the process. These variables are normally related to material and control inputs of the process. On the other hand, the output variables are influenced by both the input and intermediate variables of the process, and are normally related to the end products and the control outputs of the process. The intermediate variables measure the performance of the operating units and govern the internal state of the process. In this study, only the input and output variables are monitored and the fault conditions are introduced to the intermediate variables.

In the first-stage diagnosis, the modified ICA is used to detect and diagnose the abnormality of the process using the monitored variables. The modified ICA preserves the ranking of the PCs in the PCA whitening step and applies the same ranking on ICs, thereby important ICs can be selected for efficient monitoring of the process.[15] Suppose that a set of measurements containing n observations of d monitored variables is collected

from a normal operating process, $\mathbf{X} \in \mathbb{R}^{d \times n}$. In the original ICA, eigenvalue decomposition is applied to the covariance matrix of X to determine eigenvalues and PCs.

$$E\left[\mathbf{X}\mathbf{X}^{\mathbf{T}}\right] = \mathbf{V}\mathbf{D}\mathbf{V}^{\mathbf{T}} \tag{5.13}$$

The data samples $\mathbf{x} \in \mathbf{X}$ is then projected into the PCA subspace by computing the following matrix product.

$$\mathbf{t} = \mathbf{V}^{\mathbf{T}}\mathbf{x} \tag{5.14}$$

where $\mathbf{t}$ is the PC score vector which satisfies $E[\mathbf{t}\mathbf{t}^{\mathbf{T}}] = \mathbf{D}$. Equation (5.14) is then normalized as following:

$$\mathbf{z} = \mathbf{Q}\mathbf{x} \tag{5.15}$$

where $\mathbf{Q} = \mathbf{D}^{-1/2}\mathbf{V}^{\mathbf{T}}$ is the whitening matrix and $\mathbf{z}$ is the whitened score vector, satisfying $E[\mathbf{z}\mathbf{z}^{\mathbf{T}}] = \mathbf{I}$. Therefore, the pairwise dependence of the score vector is removed. However, it is readily observed that if FastICA algorithm is applied to further remove the high-order statistics of $\mathbf{z}$, the variance of all ICs are equal to 1 which indicates all the ICs are equally important.

$$\begin{aligned} \mathbf{s} &= \mathbf{W}\mathbf{z} \\ E[\mathbf{s}\mathbf{s}^{\mathbf{T}}] &= \mathbf{I} \end{aligned} \tag{5.16}$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$ is the normalized demixing matrix and $\mathbf{s} \in \mathbb{R}^{d \times n}$ represents the ICs. The modified ICA algorithm introduces an additional pre-processing step on the whitened score vector **z.**

$$\mathbf{y} = \mathbf{C}^{\mathbf{T}}\mathbf{z} \tag{5.17}$$

where $\mathbf{C}^{\mathbf{T}}\mathbf{C} = \mathbf{D}$. In this regard, the determined ICs have the same variance as the PCs in the whitening step; therefore the importance of these ICs can be ranked according to their variance.

$$\begin{aligned} \mathbf{s} &= \mathbf{W}\mathbf{y} \\ E[\mathbf{s}\mathbf{s}^{\mathbf{T}}] &= \mathbf{D} \end{aligned} \tag{5.18}$$

The normalized ICs is then defined as

$$\mathbf{s_n} = \mathbf{W}\mathbf{D}^{-1/2}\mathbf{y} = \mathbf{W}\mathbf{D}^{-1/2}\mathbf{C}^{\mathbf{T}}\mathbf{z} = \mathbf{W_n^T}\mathbf{z} \tag{5.19}$$

It is evident that $\mathbf{W_n^T W_n = I}$ and $E[\mathbf{s_n s_n^T}] = \mathbf{I}$. The FastICA algorithm is then applied to determine the $\mathbf{W_n}$ that maximizes the non-Gaussianity and independence of $\mathbf{s_n}$. The initial value of the demixing matrix is set as

$$\mathbf{W_n^T} = \mathbf{I}^{d\times d} \tag{5.20}$$

which is equivalent to set initial estimate of ICs $\mathbf{s_n}$ to be the whiten PCs $\mathbf{z}$. This is better than random initialization as the second-order dependence has been removed, and therefore a more consistent solution can be obtained. The detailed procedures for estimation of $\mathbf{W_n}$ using FastICA algorithm can be found in the work of Hyvärinen, Oja[73].

After the demixing matrix $\mathbf{W_n}$ for processed data vector $\mathbf{y}$ is estimated, the demixing matrix for the new online data vector $\mathbf{x^*}$ is reconstructed as following, which projects $\mathbf{x^*}$ into an ICA subspace having the same dimensionality as $\mathbf{x^*}$; in other words, all the ICs are retained.

$$\mathbf{s_n = W_x x^*} = \left(\mathbf{W_n^T D^{-1/2} V^T}\right)\mathbf{x^*} \tag{5.21}$$

$$\mathbf{A_x} = \mathbf{V D^{1/2} W_n} \tag{5.22}$$

where $\mathbf{A_x}$ is the mixing matrix, $\mathbf{A_x = W_x^{-1}}$. Since the ICs are ranked according to their variance, the dimensionality reduction through this projection can be achieved by retaining only p ($p<d$) most important ICs which capture more than 80% of variance.

$$\mathbf{s_{np} = W_{xp} x^*} = \left(\mathbf{W_{np}^T D^{-1/2} V^T}\right)\mathbf{x^*} \tag{5.23}$$

$$\mathbf{A_{xp} = V D^{1/2} W_{np}} \tag{5.24}$$

where $\mathbf{W_{np}} \in \mathbb{R}^{d\times p}$ and $\mathbf{W_{xp}} \in \mathbb{R}^{p\times d}$ contain only first p column vectors and first p row vectors of $\mathbf{W_n}$ and $\mathbf{W_x}$ respectively. Similarly $\mathbf{A_{xp}}$ contains only the first p column vectors of $\mathbf{A_x}$. It is noted that $\mathbf{A_{xp} \neq W_{xp}^{-1}}$. $\mathbf{s_{np}} \in \mathbb{R}^{p\times n}$ are the ICs with reduced dimensionality. The I^2 statistics is then computed based on these ICs.

$$I^2 = \mathbf{s_{np}^T D_p^{-1} s_{np}} \tag{5.25}$$

where $\mathbf{D_p}$ is the diagonal matrix containing first p eigenvalues of $\mathbf{D}$. The SPE statistics is computed as

$$Q = \mathbf{r^T r = (x^* - \hat{x})^T (x^* - \hat{x})} \tag{5.26}$$

$\mathbf{\hat{x}}$ is the reconstructed data vector and is given as

$$\mathbf{\hat{x} = A_{xp} W_{xp} x^*} \tag{5.27}$$

Since the ICs are non-Gaussian, for fault detection, the upper control limits for both I^2 statistics and SPE statistics under the confidence interval (1-α) × 100 are estimated by kernel density estimation.[62,103,109]

$$\hat{f}_h(I^2)=\frac{1}{nh}\sum_{i=1}^{n}K(\frac{I_i^2-E[I^2]}{h}),\quad \hat{f}_h(Q)=\frac{1}{nh}\sum_{i=1}^{n}K(\frac{Q_i-E[Q]}{h}) \tag{5.28}$$

$$\int_{-\infty}^{I_{UCL}^2}\hat{f}_h(I_i^2)dI_i^2=1-\alpha,\quad \int_{-\infty}^{Q_{UCL}}\hat{f}_h(Q_i)dQ_i=1-\alpha \tag{5.29}$$

Where h is the bandwidth and $K(\cdot)$ is the Gaussian density function, I_{UCL}^2 is the upper control limit for I^2 statistics and Q_{UCL} is the upper control limit for SPE statistics. The optimal bandwidth is determined by adopting a diffusion-based plug-in selection method [62]. For fault diagnosis, the out-of-control I^2 statistics is decomposed as following [110]:

$$\begin{aligned} I^2 &= \mathbf{s_{np}^T D_p^{-1} s_{np}} = \mathbf{s_{np}^T D_p^{-1} W_{xp} x} \\ &= \mathbf{s_{np}^T D_p^{-1}}\sum_{i=1}^{d}\mathbf{w_{xp,i}}x_i = \sum_{i=1}^{d}\mathbf{s_{np}^T D_p^{-1} w_{xp,i}}x_i \\ &= \sum_{i=1}^{d}c_i(I^2) \end{aligned} \tag{5.30}$$

Where $c_i(I^2)=\mathbf{s_{np}^T D_p^{-1} w_{xp,i}}x_i$ is the contribution of the i[th] monitored variables. $\mathbf{w_{xp,i}}$ is the i[th] row vector of $\mathbf{W_{xp}}$ and x_i is the i[th] entry of $\mathbf{x}$. Alternatively, the SPE contribution is calculated as following:

$$c_i(SPE)=r_i^2 \tag{5.31}$$

Where e_i is the i[th] entry of $\mathbf{r = x - \hat{x}}$ representing the contribution of the i[th] monitored variable to the residual. Finally, the faulty monitored variables is identified to be the one having the highest contribution based on $c_i(I^2)$.

$$x_{root}=\arg\max_{x_i}\left(\mathbf{s_{np}^T D_p^{-1} w_{xp,i}}x_i\right) \tag{5.32}$$

Subsequently, the following evidence is generated at the corresponding variable node in BN for the second-stage diagnosis of the true root-cause.

$$\begin{aligned} &e\rightarrow P(x_{root}=s_0)=1, s_0\in\{S\} \\ &S=\begin{cases} s_0 & \text{If fault} \\ s_1 & \text{If Normal} \end{cases} \end{aligned} \tag{5.33}$$

5.3.2 Bayesian Network for Second-stage Fault Diagnosis

BN models the input variables, intermediate variables and output variables as the root nodes, intermediate nodes and leaf nodes respectively. To determine the network structure, the hierarchy and causal dependence amongst these nodes are extracted from the process flow diagram. The process flow diagram indicates the general flow of operation from upstream operating units to downstream operating units and also provides information of the physical and chemical interaction between the process variables. The hierarchy of the network is determined by first associating the nodes with each operating unit and then arranging them according to the process flow order. The causal dependence amongst these nodes is identified by analysing the physical and chemical interaction between process variables. In case of process with recycle loop in which downstream process variables are fed back to the upstream variables, two nodes are created for this process variable in BN to capture the recycling feature, with one node representing the process variable and another dummy node representing the recycled process variable. This special condition, as illustrated in Figure 5-2, is necessary to satisfy the acyclic characteristic of BN.

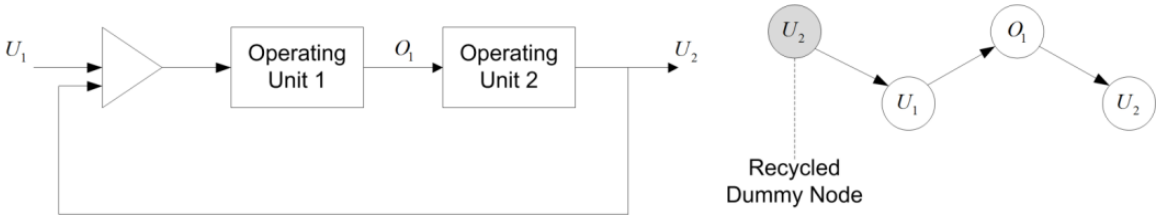

Figure 5-2: Special condition for recycled process variable.

Once the BN structure is determined, the parameters which describe the statistical dependence (CPT) amongst the nodes are estimated from process normal operating data through Eqs (5.11) and (5.12). The BN is now ready for second-stage diagnosis. The second-stage diagnosis consists of three steps: (1) updating the state of all nodes in BN given the evidence determined by the Modified ICA in the first-diagnosis stage; (2) determine the most probable explanation (MPE) of the evidence and identify the most probable set of faulty process variables; (3) locate the true root-cause variable as the one within the most probable set and having the highest updated probability of fault. The sum-product algorithm is used in the first step to calculate the posterior probability of the nodes in each state given the evidence. The sum-product algorithm is a general form of the forward-backward algorithm.[111] It computes marginal distribution of a target node in a factor graph through passing messages inward from an arbitrary set of nodes at the edge of the factor graph and combining them at the target node. The details of the sum-product algorithm are presented in the Appendix.

After the posterior probability of each node is computed by sum-product algorithm, the max-product algorithm with back-tracking is used to identify the most likely state of each node. The set of nodes that have been identified to be most likely in faulty state, s_0, represents the most probable set of faulty process variables which contribute significantly to the faulty state of the identified faulty monitored variable. The max-product algorithm

is a generalization of the Viterbi algorithm on graphical model.[112] The procedure of max-product algorithm is similar to sum-product algorithm except that the sum-out operator is replaced by a max-out operator.[113] The detailed procedures of the max-product algorithm are presented in Section 9.10 of the Appendices. The use of messages in the sum-product and max-product algorithm allows efficient computation of marginal (initial) and posterior probability of each node. For example, as shown in Figure 9-3, the messages are only computed once for each node and are accumulated at the factorial nodes. As a result, the message arriving at the next node already contains information from all the other nodes in the path. In this respect, the computational complexity scales polynomial in the number of nodes. It is much more efficient than brute-force search which calculates and compares the probability of all possible combinations of states of the nodes to determine the marginal probability. In fact, the computational complexity of the brute-force search scales exponentially in total number of nodes.

It has to be noted that some of the variables included in the most probable set may have lower probability of fault than those that are determined to be most likely normal. The reason is as follow. Considering the variable node x_j in Figure 9-3 which is assumed to be one of the identified faulty nodes, it is also assumed that the message passed from x_1 carries information indicating x_1 may have a high probability of fault $\mu_{x_1 \to f_{i-j}}(x_1 = s_0) = 0.9$, and also the rest of input variables $x_{1:n}$ all pass messages that have very low probability of fault $\mu_{x_i \to f_{i-j}}(x_i = s_0) = 0.1, i \in \{2,3,4,...,n\}$. As a result, the combined message to x_j is equal to $0.9 \times 0.1^{n-1}$ which has a very low probability of fault, and therefore upon updating, the probability of fault of x_j is substantially lower though x_j is still included in the most probable set. The step-by-step procedure of the proposed two-stage fault diagnosis technique is illustrated as a logic flow diagram in Figure 5-3.

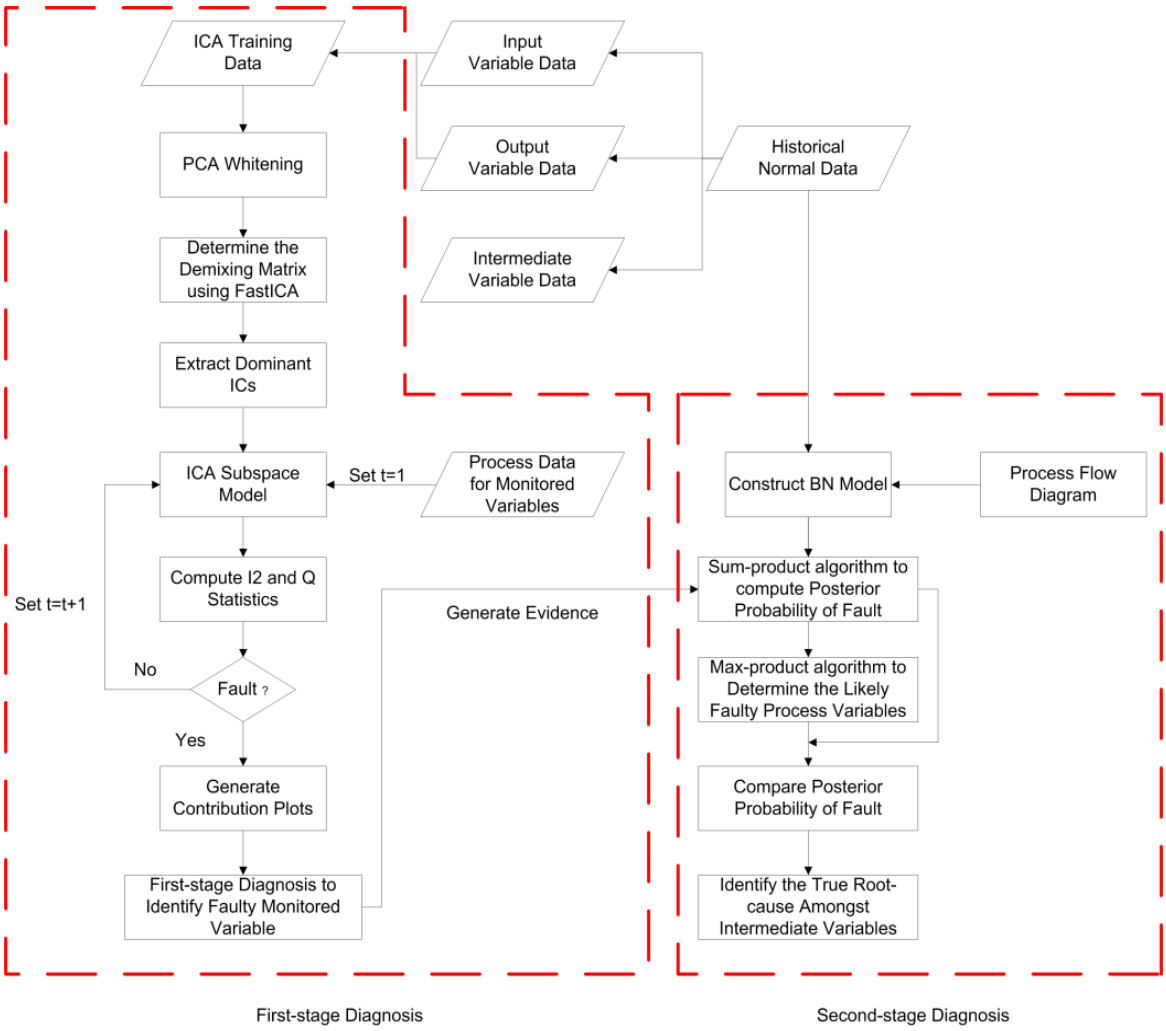

Figure 5-3: Flow diagram of the proposed two-stage fault diagnosis.

5.4 Case Studies

5.4.1 A simple multivariate process

The effectiveness of the two-stage fault diagnosis technique is first demonstrated on a simple multivariate process which has been proposed by Lee, et al.[1] The state-space model of this multivariate process is shown as following:

$$\mathbf{z}(i) = \begin{bmatrix} 0.018 & -0.191 & 0.287 \\ 0.847 & 0.264 & 0.943 \\ -0.333 & 0.514 & -0.217 \end{bmatrix} \mathbf{z}(i-1) + \begin{bmatrix} 1 & 2 \\ 3 & -4 \\ -2 & 1 \end{bmatrix} \mathbf{u}(i-1)$$

$$\mathbf{y}(i) = \mathbf{z}(i) + \mathbf{v}(i)$$

$$\mathbf{u}(i) = \begin{bmatrix} 0.811 & -0.226 \\ 0.447 & 0.415 \end{bmatrix} \mathbf{u}(i-1) + \begin{bmatrix} 0.193 & 0.689 \\ -0.320 & -0.749 \end{bmatrix} \mathbf{h}(i-1)$$

The input vector $\mathbf{u} = [u_1 \quad u_2]^T$ consists of two variables which are affected by an external disturbance $\mathbf{h} = [h_1 \quad h_2]^T$. Each entry of $\mathbf{h}$ is uniformly sampled from the

interval [-2, 2]. The output vector $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & y_3 \end{bmatrix}^T$ has three variables which are dependent on $\mathbf{z} = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T$ and a random noise vector $\mathbf{v}$ with zero mean and a covariance of $0.1 \times \mathbf{I}^{3\times 3}$. In this example, the input and output variables form the set of monitored variables, $\mathbf{x} = \begin{bmatrix} u_1 & u_2 & y_1 & y_2 & y_3 \end{bmatrix}$, and $\mathbf{z} = \begin{bmatrix} z_1 & z_2 & z_3 \end{bmatrix}^T$ is chosen to be set of intermediate variables. Two hundred normal data samples have been generated for determining the ICA subspace and training the BN. During process operation, two fault conditions are introduced as step changes (magnitude of 5) to z_1 and z_3 respectively, at sample interval 200. For first-stage diagnosis, the first three ICs that capture 90% variance are selected to build the ICA subspace. The process monitoring results based on the modified ICA for both fault conditions are shown in Figure 5-4 and Figure 5-5 respectively. The I^2 statistics are able to capture the abnormal behaviour of process operation after sample number 200 for both fault cases (after fault is introduced, almost all data points breach the upper control limit). In contrast, the SPE statistics yields unsatisfactory performance for both cases as a significant amount of data points still fall under the upper control limit even in the presence of a fault condition. The explanation for these observations is as follow. Due to the complex interactions between the variables in the state-space model, the monitored data contains significant non-Gaussian features. These non-Gaussian features are retained in the ICA subspace and are measured by the I^2 statistics. On the other hand, the remaining Gaussian features of the monitored data are retained in the residual space and are measured by the SPE statistics, which leads to the poor performance of SPE statistics. Nevertheless, in the contribution plots (I^2 and SPE), the monitored variables *y1* and *y3* have the highest contribution to fault condition 1 and fault condition 2, respectively. It is observed that because of the lack of online monitored data of process variables z_1, z_2, and z_3, the modified ICA is not able to provide diagnosis results on these three variables.

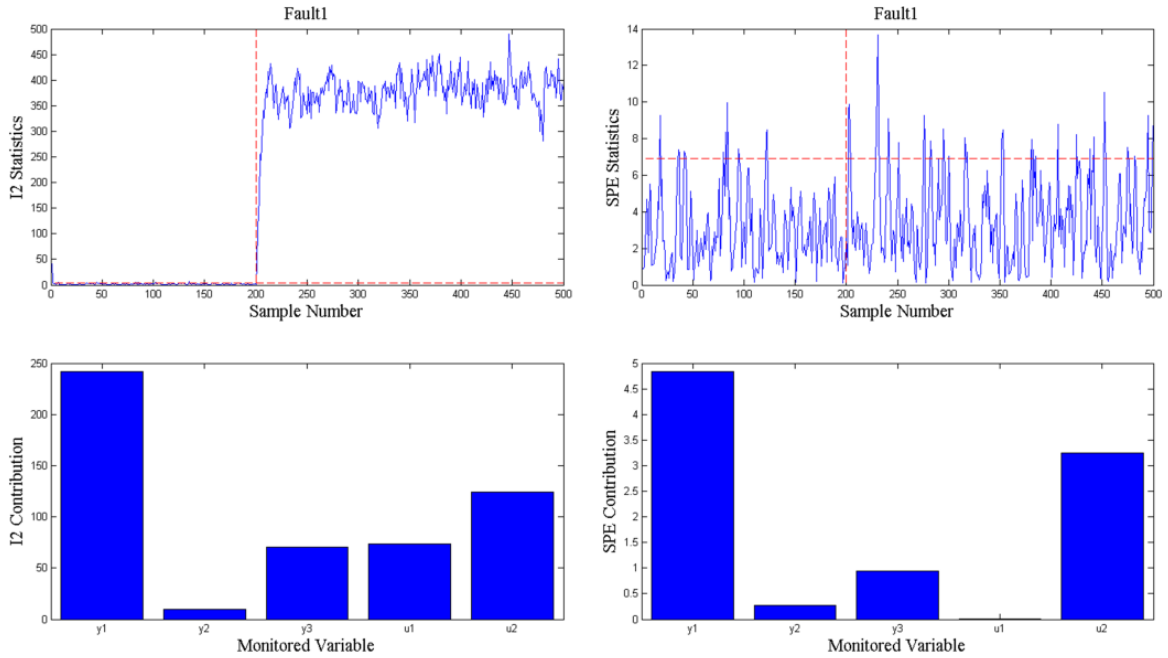

Figure 5-4: Modified ICA process monitoring results for fault condition 1.

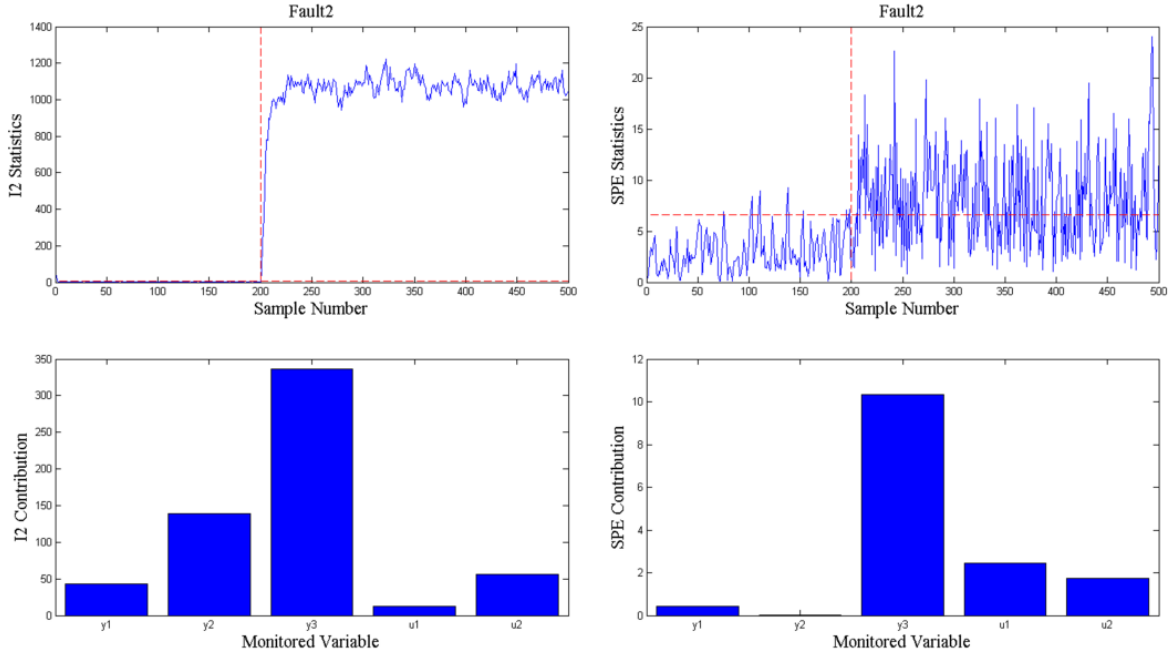

Figure 5-5: Modified ICA process monitoring results for fault condition 2.

In the second-stage diagnosis, the process variables are first arranged according to the order of process flow: $\mathbf{h} \rightarrow \mathbf{u} \rightarrow \mathbf{z} \rightarrow \mathbf{y}$. Then, the causal relationships amongst these variables are determined from the state-space model and the CPDFs are estimated from the normal process data using the method outline in section 5.2.2. As the normal process data is real-valued, it has to be discretised for the purpose of training the network with discrete states. In this case, the GeNIe Bayesian network package [114] is used to hierarchically discretise the normal process data and train the network. Figure 5-6 shows the trained BN model and the initial probability distribution over the initial states (s_0 for

fault and s_1 for normal) of each node. As an example, the estimated CPDFs of node u1 is summarized as a conditional probability table (CPT) shown in Table 5-1. The CPDFs in the CPT describe the likelihood of a given state of the ancestor nodes leading to a particular state of the descendent node. For instance, in the first column of the Table 5-1, when both h1 and h2 are in state 0, the likelihood of u1 in state 0 is 0.844 and the likelihood of u1 in state 1 is 0.156. It is noted that these two values sum up to 1.

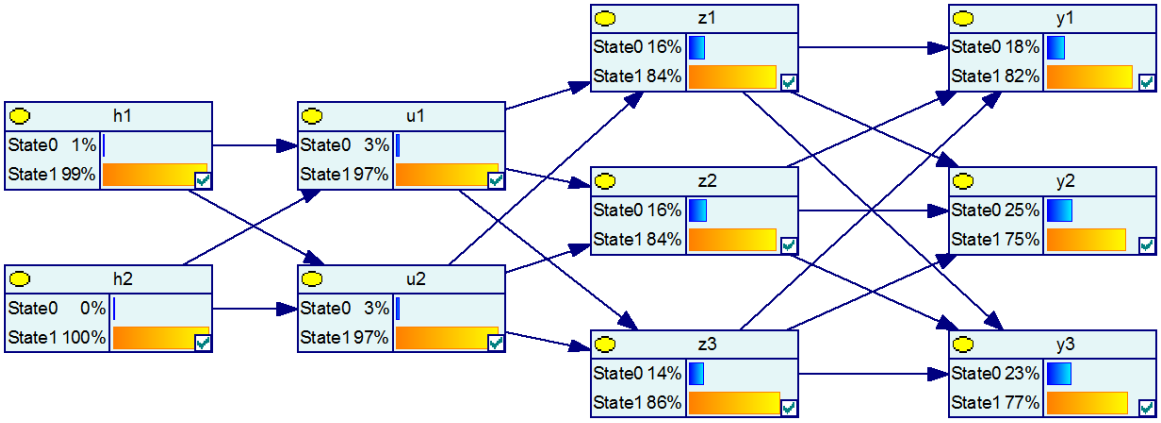

Figure 5-6: BN for the simple multivariate process with initial probability distributions.

Table 5-1: Conditional Probability Table for node u1.

h1	State 0		State 1	
h2	State 0	State 1	State 0	State 1
State 0	0.844	0.908	0.912	0.020
State 1	0.156	0.092	0.088	0.980

In the first fault condition, y_1 has been identified as the faulty monitored variable. Evidence is then generated at the corresponding node in BN and the posterior probability distribution of each node is obtained through sum-product algorithm. In addition, the most probable set of faulty process variables is determined through using the max-product algorithm with back-tracking. As shown in Figure 5-7, the evidence node is enclosed by a red rectangle and the set of fault nodes are enclosed by blue rectangles. In comparison, the intermediate variable z_1 has not only the highest probability of fault (51%) but also the highest percentage increase in probability of fault (35%) amongst the most probable set; z_1 is correctly identified as the true root-cause variable. The second-stage diagnosis results for the second fault condition are shown in Figure 5-8. Similarly, the intermediate process variable z_3 has the highest probability of fault (42%) and percentage increase (28%) in probability amongst the most probable set, and therefore it has also been correctly identified as the true root-cause. It is observed that u_2 with low probability of fault is also included in the most probable set. This variable is in the direct path of the maximum message passing; however, due to the low posterior probability of fault of h_1 and h_2, as explained in the last paragraph of section 5.3.2, the product of the messages arriving at u_1 has a very low probability of fault, therefore resulting in low posterior probability of fault of u_2. As demonstrated by this simple example, the modified ICA

based technique alone is not able to locate the true root-cause if the process variable is not monitored and the online process data is not available. In contrast, the proposed two-stage diagnosis technique conducts further probabilistic reasoning on the results obtained by the modified ICA using BN and is capable of locating the not monitored true root-cause intermediate variable.

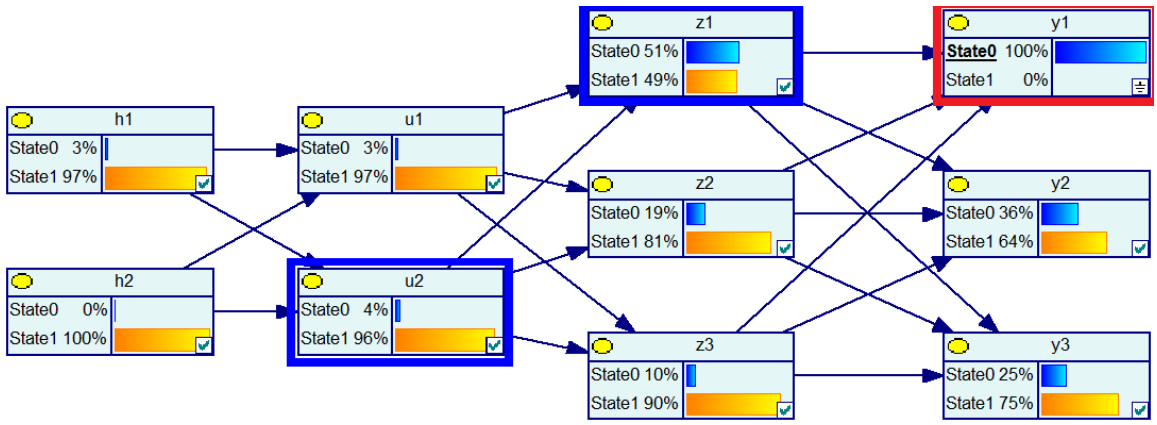

Figure 5-7 Second-stage fault diagnosis results for fault condition 1

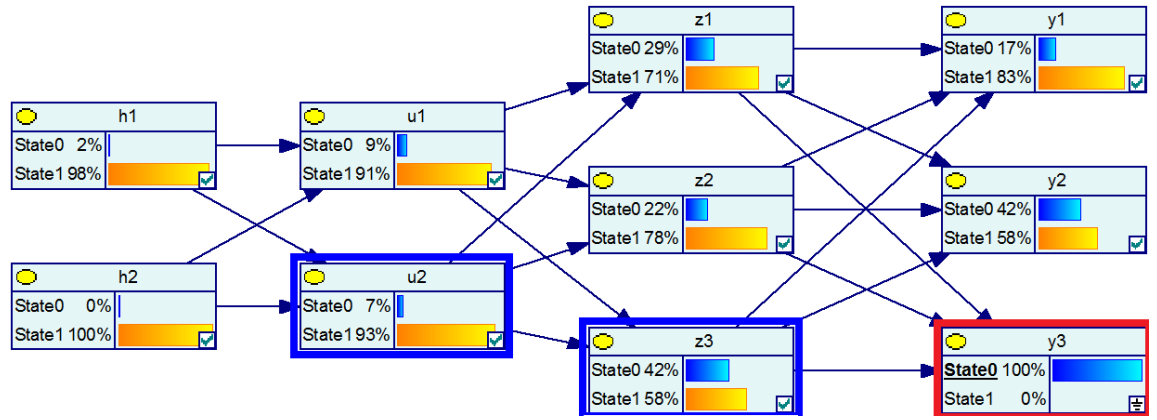

Figure 5-8: Second-stage fault diagnosis results for fault condition 2

5.4.2 Tennessee Eastman Chemical Process

In this section, the effectiveness of the proposed two-stage fault diagnosis technique is verified by further testing on a well-established Simulink program of the Tennessee Eastman chemical process. The Simulink program adopts the decentralized control strategy to construct a closed-loop stable simulation of the process.[86] The Tennessee Eastman chemical process consists of five major operating units: a reactor, a product condenser, a vapour-liquid separator, a recycle compressor, and a product stripper.[32] The process flow diagram of the process system is shown in Figure 9-1.

In total, 41 process variables are measured for the process system, among which 22 process variables are monitored to determine the operating condition of the process system. These monitored variables are listed in Table 9-1.[32] In addition, Table 9-2 summarizes the 20 fault conditions that have been widely used in process monitoring research as base cases for comparison of various approaches.[32]

In this case study, the listed 22 process variables are further classified into input variables, intermediate variables and output variables. The input variables are associated with material and control input to the process (C1, C2, C3, C4, C5, and C19). The output variables are those used to determine the quality of the end product and also include the control output (C5, C8, C10, C17, C18, C21, and C22). The rest of the process variables are intermediate variables. It is noted that C5 is a recycled variable. To capture this loop feature, C5 has been included in both input and output variable group. The input variables and output variables are combined to form the monitored variable set, while the intermediate variables are not monitored. One thousand normal data samples are collected for obtaining the ICA subspace model the BN model. Three fault conditions including IDV11, IDV12 and the compressor recycle valve sticking have been generated for testing, at sample interval 3000. These three faults are selected as they are not directly related to the monitored variables; that is, these fault conditions are originated from the not monitored intermediate variables. The true root-cause process variables for each of the introduced fault condition are listed in Table 5-2.

Table 5-2: True root-cause process variables for the introduced fault conditions.

Fault ID.	True root-cause process variable
IDV11	C9, Reactor Temperature
IDV12	C11, Separator Temperature
Compressor recycle valve sticking	C20, Compressor Work

For first-stage fault detection, the first 9 ICs that capture 80% of the variance are used to construct the ICA subspace model. The modified ICA process monitoring results for all three cases are shown in Figure 5-9, Figure 5-10, and Figure 5-11, respectively. In general, I^2 statistics provides better fault detection results as compared to SPE statistics due to the reason that I^2 statistics captures non-Gaussian feature while SPE statistics only captures the residual Gaussian features. On the other hand, for IDV11 and the compressor recycle valve fault, both statistics are able to locate the most closely related monitored variables C21 (Reactor cooling water outlet temperature) and C5 (Recycle flow). However, for IDV12, the SPE statistics contribution fails to locate the faulty monitored variable C18 (Stripper temperature). It is readily observed that the identified faulty monitored process variables are not the true root-cause variable. The modified ICA is not able to isolate the true root-cause from the intermediate variables due to the lack of online monitored data.

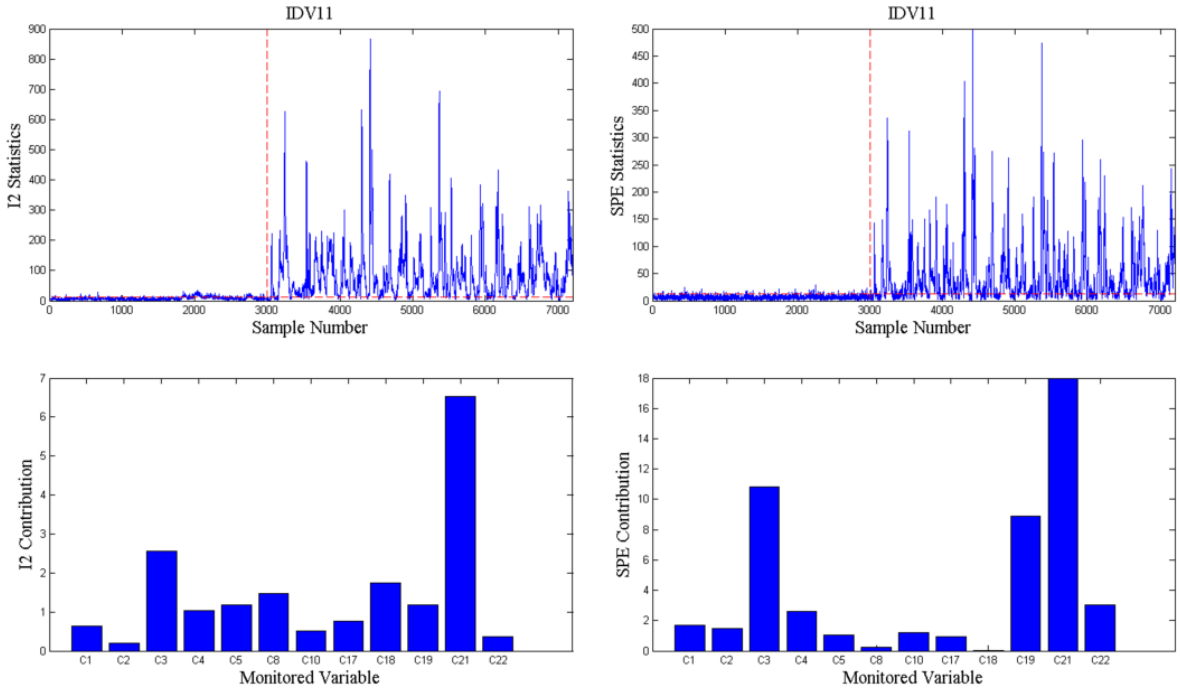

Figure 5-9: Modified ICA based process monitoring results for IDV11.

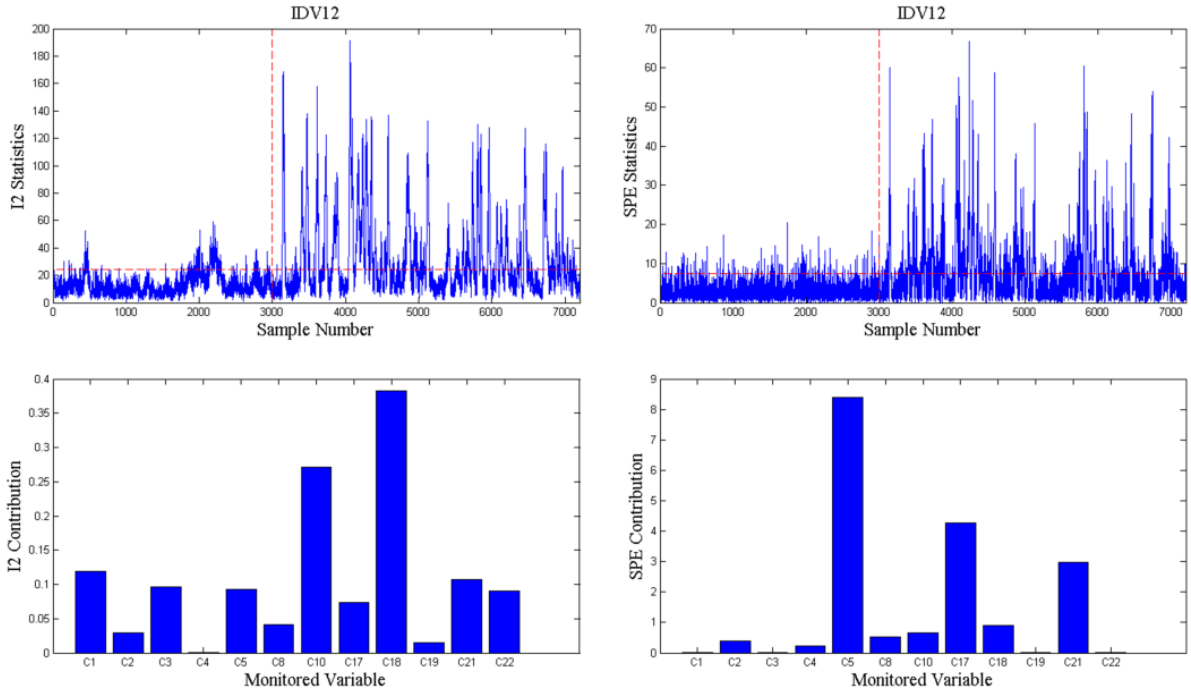

Figure 5-10: Modified ICA based process monitoring results for IDV12.

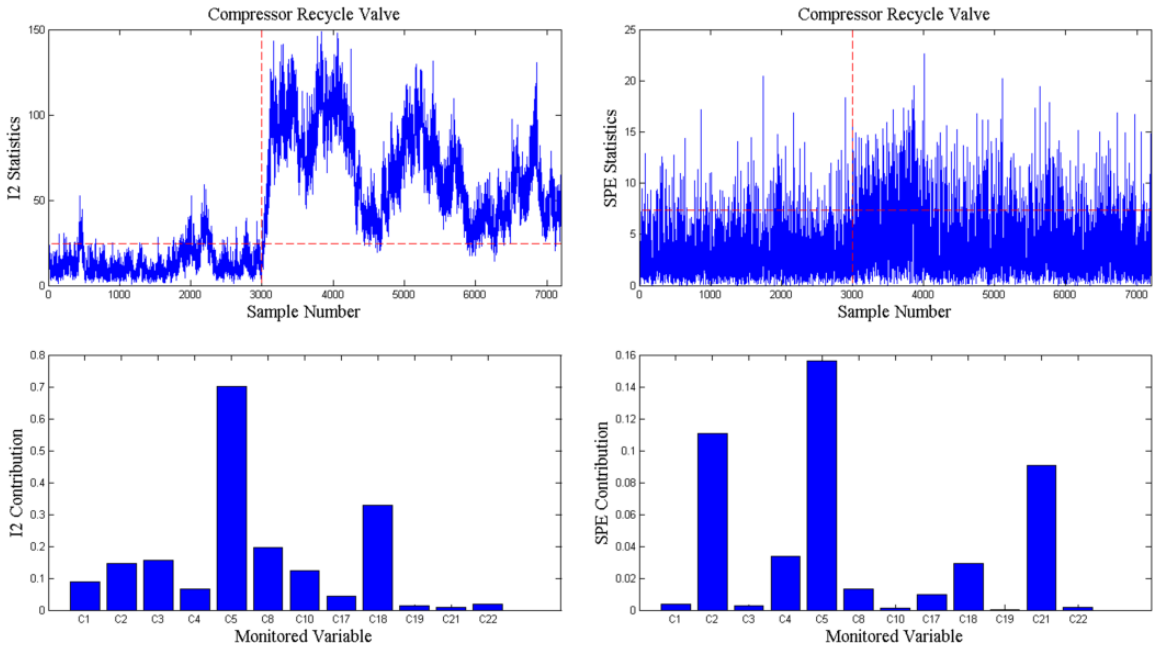

Figure 5-11: Modified ICA based process monitoring results for compressor recycle valve.

In the second-stage fault diagnosis, process variables are related to each operating unit and are arranged according to the process flow order in **Error! Reference source not ound.**. A dummy node is created for C5 to represent the recycle loop. The causal dependence amongst these nodes is obtained by logically analysing the chemical and physical interactions between the process variables. The Bayesian network parameters (CPDFs) are estimated from the historical normal data which covers all process variables using the GeNIe BN package. The determined BN model and the initial condition of each node are shown in Figure 5-12. In addition, Table 5-3 summarizes the conditional probability table of node C6 (Reactor feed rate). Similar to the Table 5-1 of the first case study, the conditional probabilities (CPDFs) in Table 5-3 provide the likelihood of different combinations of states of C6's ancestor nodes leading to either faulty or normal state of C6.

Table 5-3: Conditional Probability Table for node C6 (Reactor feed rate)

C1	**State 0**								**State 1**							
C2	**State 0**				**State 1**				**State 0**				**State 1**			
C3	**State 0**		**State 1**		**State 0**		**State 1**		**State 0**		**State 1**		**State 0**		**State 1**	
C5_1	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**	**0**	**1**
State 0	0.084	0.044	0.114	0.042	0.05	0.357	0.05	0.05	0.112	0.051	0.208	0.039	0.05	0.05	0.05	0.041
State 1	0.916	0.956	0.886	0.958	0.95	0.643	0.95	0.95	0.888	0.949	0.792	0.961	0.95	0.95	0.95	0.959

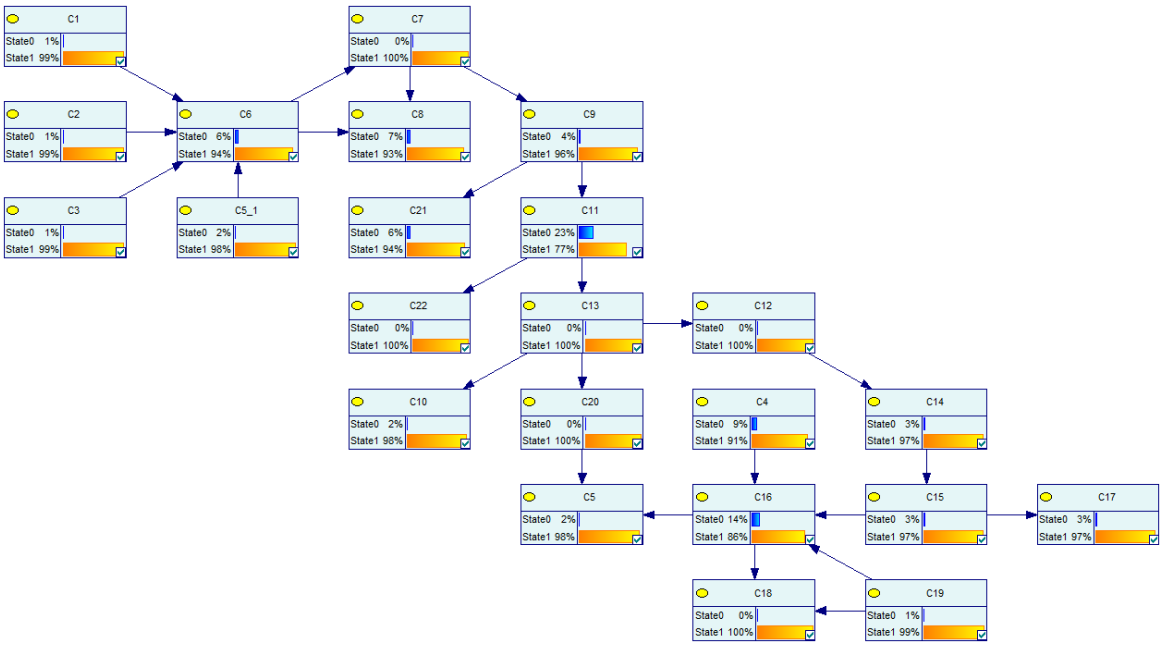

Figure 5-12: BN with initial condition for the Tennessee Eastman chemical process.

In IDV11, the monitored variable C21 has been identified to be faulty in the monitored set. The evidence is generated at the corresponding node (C21) in BN to set the probability of fault equal to 1. Sum-product algorithm is then used to obtain the posterior probability distribution of all the nodes given this evidence. Subsequently, the max-product algorithm with back-tracking is applied to identify the most probable set of faulty nodes. The second-stage diagnosis results for IDV11 are shown in Figure 5-13, with the red rectangle enclosing the evidence and blue rectangle enclosing the most probable set of faulty nodes. The set of faulty nodes also shows the fault propagation path. The comparison of probability of fault of the identified set of faulty variables before and after updating along with the fault propagation path is shown in Figure 5-14. It is readily observed that both C9 and C11 have very high probability of fault (68% and 69%). However, since C11 has a much higher initial probability of fault, its percentage increase in probability of fault is less than C9, which means C9 is more likely to be in a faulty state. The increased random variation in the reactor cooling water inlet temperature has direct effect on the reactor temperature (C9). The abnormal behaviour of the reactor temperature (C9) causes the deviation of the separator temperature from its normal state. This fault effect is then further propagated to upset the separator pressure (C13), separator level (C12) and separator underflow (C14). Eventually, because of the undesired variation of the separator flow, the stripper level (C15) and stripper pressure (C16) are adversely affected. In this regard, C9 has been correctly identified as the true root-cause amongst the intermediate variables which are not monitored and the fault propagation path coincides well with the logical analysis of process flow.

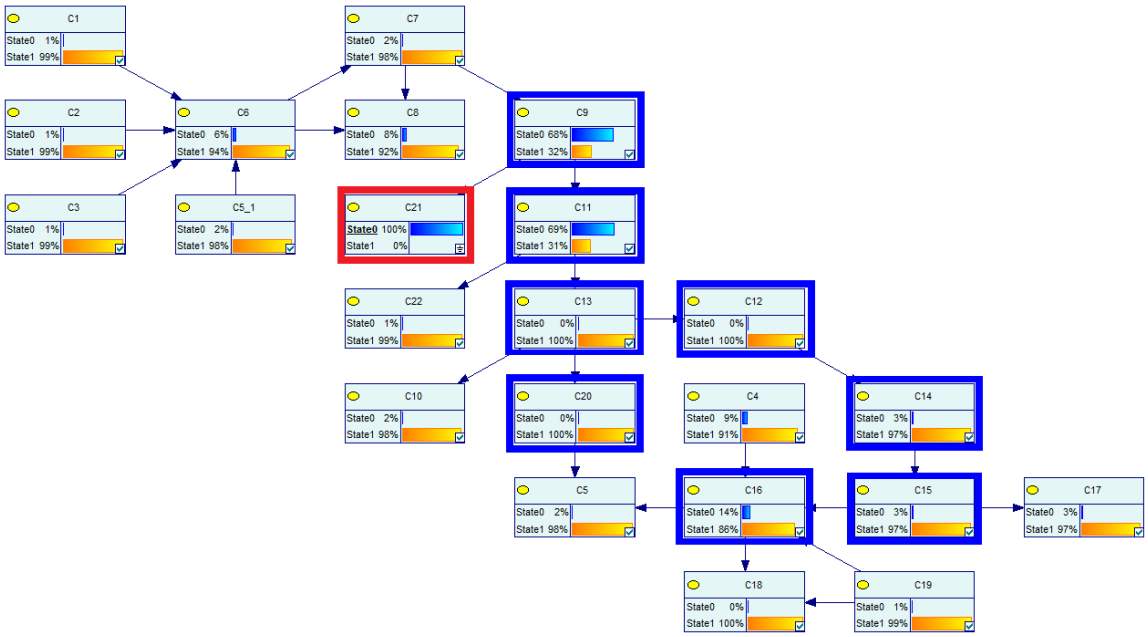

Figure 5-13: Second-stage fault diagnosis results for IDV11.

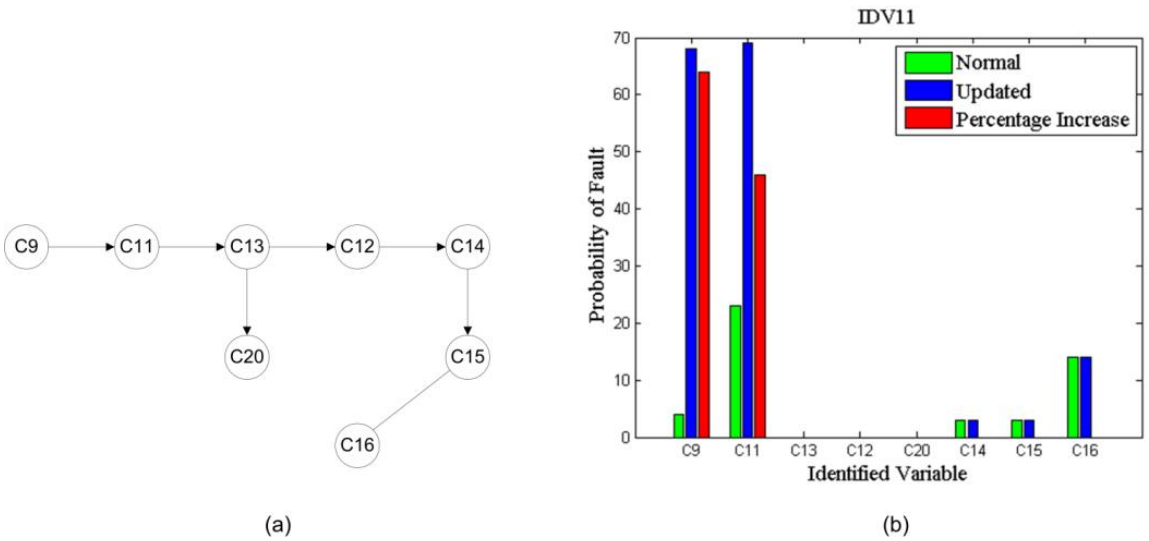

Figure 5-14: Fault propagation path and comparison of probability of fault for IDV11.

The second-stage fault diagnosis results for IDV12 are shown in Figure 5-15 and Figure 5-16. Separator temperature (C11) is identified as the true root-cause intermediate variable with 47% probability of fault and 47% increase in probability of fault. Because the abnormal variation of the condenser cooling water inlet temperature, the temperature of the product stream to the separator is deviated away from the normal level. Subsequently, the separator temperature (C11) is directly impacted and starts to exhibit abnormal behaviour which then causes abnormal variation of the separator pressure (C13). The pressure of the recycled stream is also affected, leading to change in output work of the compressor. This fault condition is then propagated to stripper unit to disrupt the stripper temperature (C18) and stripper pressure (C16). And eventually the recycle

flow (C5) is affected as a result of the combined fault condition in compressor work (C20) and the stripper pressure (C18).

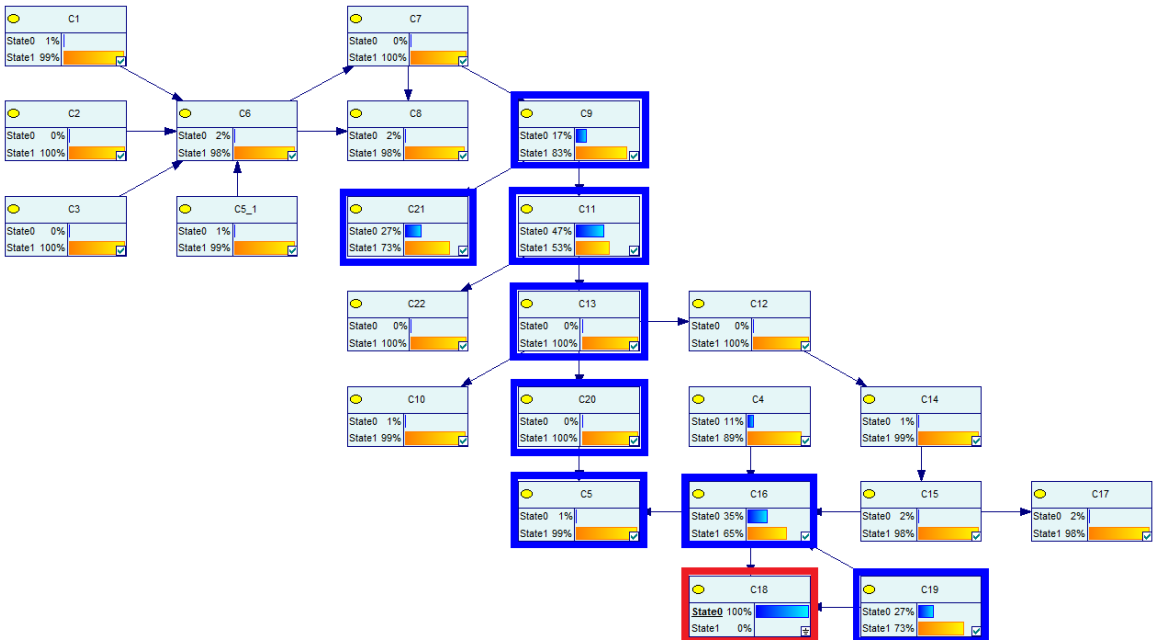

Figure 5-15: Second-stage fault diagnosis results for IDV12.

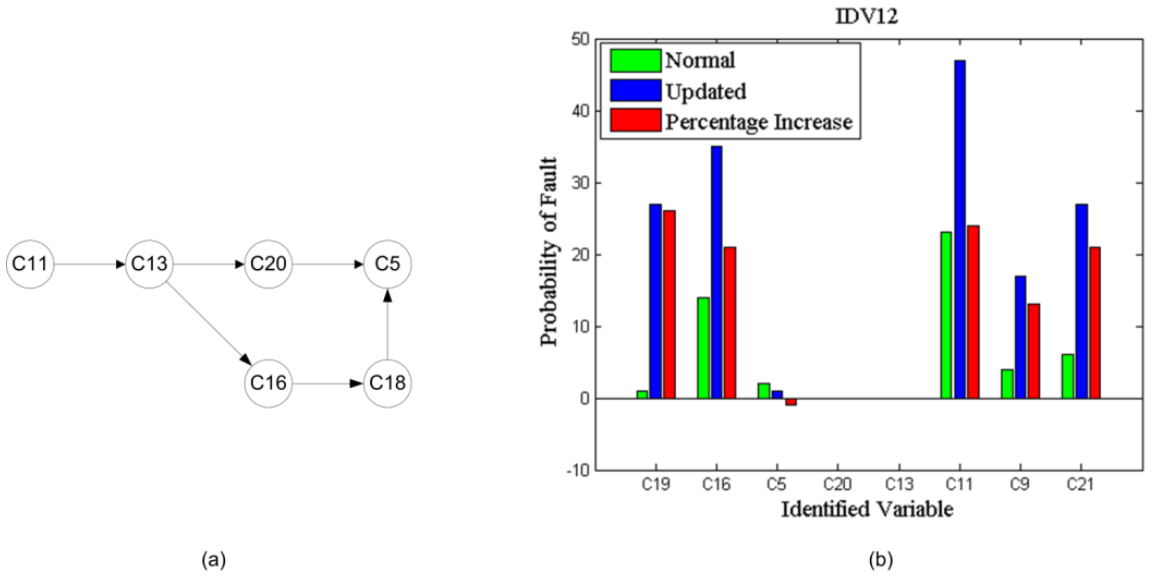

Figure 5-16: Fault propagation path and comparison of probability of fault for IDV12.

In the last case, the results for fault diagnosis are shown in Figure 5-17 and Figure 5-18. Sticking of the compressor valve forces the compressor to work in a suboptimal condition which results in reduced compressor output work. The recycle flow is also reduced. The reduced recycle flow returns back to the reactor and causes changes in the reactor feed (C6) which subsequently disrupts the dynamic balance of the chemical reaction in the reactor. This condition then affects the reactor pressure (C7), reactor temperature (C9) and further propagates downstream to cause undesired variation of

separator temperature (C11), separator pressure (C13), and eventually the stripper pressure (C16). The effect of the fault condition is deteriorated as it is circulated back to the reactor with the recycle flow. The identified most probable set of fault variables form a fault probation path that reflects well the actual behaviour of the process in fault condition. The intermediate variable C20 has also been correctly identified as the true root-cause variable with the highest posterior probability of fault (58%) and percentage increase in probability of fault (58%). In summary, the effectiveness of the proposed two-stage fault diagnosis has been successfully verified on the Tennessee Eastman chemical process. It has been demonstrated that by combining the Modified ICA and BN, the true root-cause variable can be accurately located even the online process measurement data is not available.

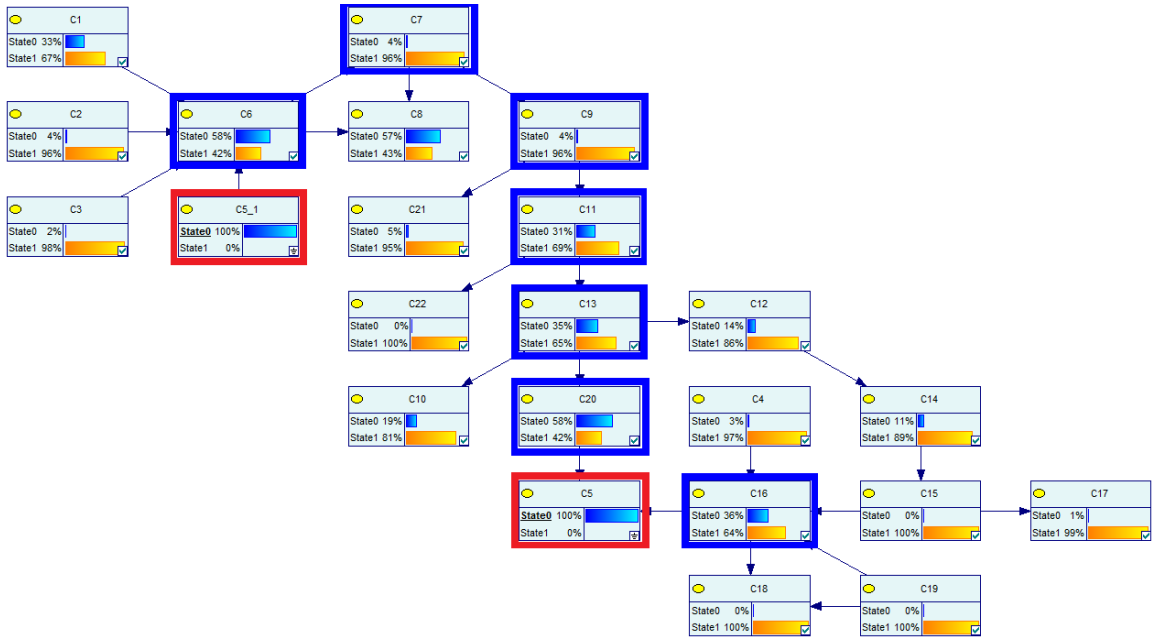

Figure 5-17: Second-stage fault diagnosis results for compressor valve.

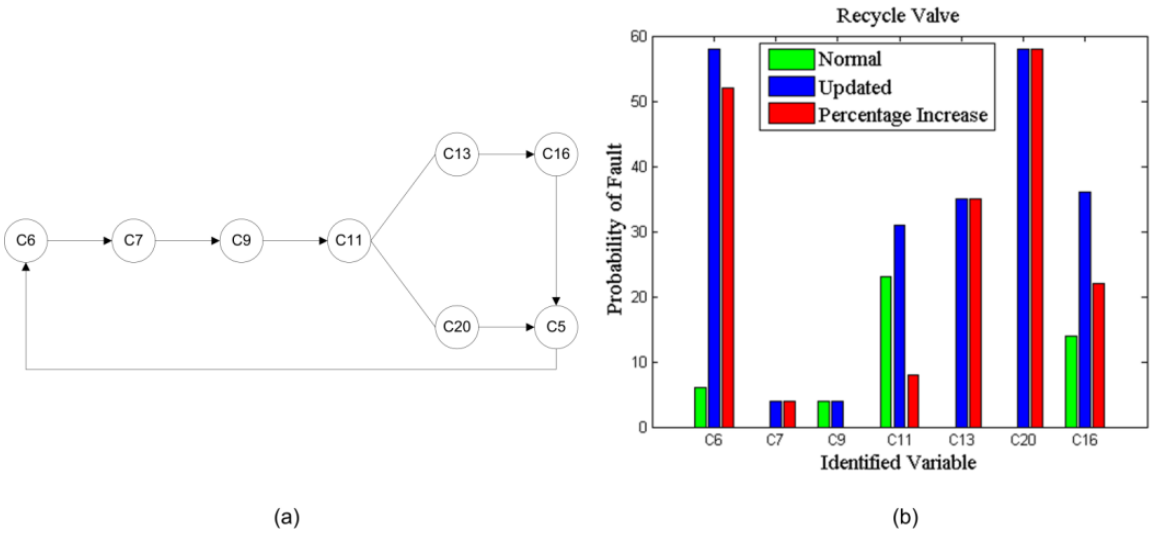

Figure 5-18: Fault propagation path and comparison of probability of fault for compressor work.

To further demonstrate the advantage of the proposed technique, the results of the case studies are also compared with diagnosis results of the modified ICA when all 22 process variables are used. The diagnosis results of IDV11, IDV12, and the compressor recycle valve fault based on the modified ICA with 22 process variables being monitored are shown in Figure 5-19. In comparison to Table 5-2, it is observed that I^2 statistics is able to accurately locate true root-cause process variables for IDV11 (C9) and IDV12 (C11); while the true root-cause for the compressor recycle valve sticking is not correctly identified. On the other hand, the SPE statistics is only able to diagnose the fault condition IDV11. The results have shown that with all 22 monitored process variables

included, the accuracy of fault diagnosis based on the modified ICA is not guaranteed. The unsatisfactory performance is possibly attributed to the increase in number of monitored variables which introduces more complex features in the monitored process data. Furthermore, with the increased dimensionality of the data space, this technique may not be able to efficiently extract the additional features to construct a robust feature space, therefore leading to poor performance. In contrast, the proposed technique requires less monitored variables to produce consistent diagnosis results which accurately identify the true root-cause process variables. In practice, the proposed technique provides the advantage of reducing the monitoring cost and minimizing the potential of false diagnoses. To better demonstrate the advantage of the proposed two-stage diagnosis technique over the conventional MICA-based technique, a comparison of the fault diagnostic performance of these two techniques is summarized in Table 5-4.

Table 5-4: Comparison of of fault diagnosis performance between the proposed technique and the MICA-based techniques

True root-cause variable	Fault diagnosed by proposed technique		Fault diagnosed by MICA-based technique, I^2 contribution		Fault diagnosed by MICA-based technique, SPE contribution	
	Correct diagnosis? [Yes/No]		Correct diagnosis? [Yes/No]		Correct diagnosis? [Yes/No]	
C9, Reactor Temperature	C9	Yes	C9	Yes	C9	Yes
C11, Separator Temperature	C11	Yes	C11	Yes	C2	No
C20, Compressor Work	C20	Yes	C9	No	C9	No

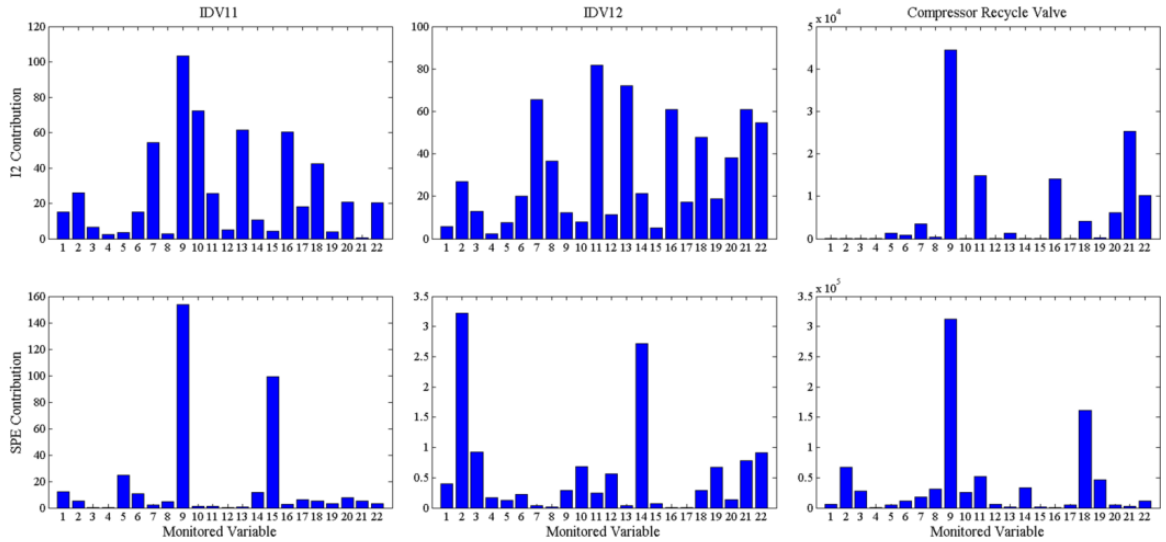

Figure 5-19: Fault diagnosis results based on modified ICA with 22 process variables

5.5 Conclusion

In this study, a Modified ICA and BN based two-stage fault diagnosis technique is developed. The proposed technique addresses the issue that the conventional statistical techniques are incapable of isolating fault from not monitored process variables. The Modified ICA is used in the first-stage diagnosis to detect and diagnose fault based on monitored process variables. In the second-stage, a Bayesian network of the process is constructed and trained to capture the qualitative and quantitative dependencies among all the process variables. Subsequently, the results from the first-stage diagnosis are used to conduct deductive and adductive reasoning on the Bayesian Network to isolate the true root-cause variable. The effectiveness of the proposed technique is verified on a simple multivariate process and the Tennessee Eastman chemical process. The results from both case studies demonstrate that the proposed technique is able to isolate the true root-cause variable from off-line process variables. In addition, by analysing the dependency structure of the Bayesian network, it is also possible to identify the most probable propagation path of the fault. The proposed technique provides a more robust process monitoring with less number of monitored variables required. This effectively reduces the monitoring cost and minimizes the chance of false diagnoses.

In future research, this work will be integrated with quantitative risk analysis approach to form a unified framework for a real-time risk management system. This risk management system is able to readily assess the operational risk and suggest optimal remedial actions or safety measures to minimize the impact of the fault on the process, therefore preventing the process from catastrophic failure.

6 Risk-based Fault Detection using Self-Organizing Map

Abstract

The complexity of modern systems is increasing rapidly and the dominating relationships among system variables have become highly non-linear. This results in difficulty in the identification of a system's operating states. In turn, this difficulty affects the sensitivity of fault detection and imposes a challenge on ensuring the safety of operation. In recent years, Self-Organizing Maps has gained popularity in system monitoring as a robust non-linear dimensionality reduction tool. Self-Organizing Map is able to capture non-linear variations of the system. Therefore, it is sensitive to the change of a system's states leading to early detection of fault. In this paper, a new approach based on Self-Organizing Map is proposed to detect and assess the risk of fault. In addition, probabilistic analysis is applied to characterize the risk of fault into different levels according to the hazard potential to enable a refined monitoring of the system. The proposed approach is applied on two experimental systems. The results from both systems have shown high sensitivity of the proposed approach in detecting and identifying the root cause of faults. The refined monitoring facilitates the determination of the risk of fault and early deployment of remedial actions and safety measures to minimize the potential impact of fault.

Keywords: Self-Organizing Map, Risk Assessment, Probabilistic Analysis, Bayesian Updating

6.1 Introduction

The rapid increase in complexity of modern systems imposes a challenge towards ensuring the safety of operations. This increase in complexity is directly related to the number of variables a system comprises. Each variable represents an individual dimension. To ensure the safety of a system, multiple variables have to be monitored simultaneously. This requires a tool with reliable high dimensionality handling capabilities. In addition, as the dimensionality increases, the relationships among system variables become highly non-linear. The identification of these non-linear relationships enables precise monitoring of behaviours of variables which is another key aspect concerning the safety of systems [115]. The disruptions of the relationships among system variables can cause abnormal behaviours which are considered as faults. The potential impact on the safety of system increases with the progression of fault. To minimize the impact, it is best to detect the fault at its early stage; this requires the development of a fault detection approach with high sensitivity. Also, the progression of fault needs to be traced to facilitate the efficient determination of safety measures and remedial actions to minimize the impact.

In many cases, the monitoring of complex systems is achieved through a technique known as dimensionality reduction. In general, variables that represent the most variances of system are combined to form a new set of variables and the variables representing less variance are disregarded. The system is then monitored based on the new set of variables which has less dimensionality.

In recent years, Self-Organizing Maps (SOMs) have gained popularity in fault detection and identification of complex systems as an efficient dimensionality reduction technique [116]. SOM has the ability of capturing nonlinear relationships of high dimensional data and visualizing them on a low-dimensional display in a topologically ordered fashion known as feature clusters.[116-119] This feature of SOM makes it sensitive to the change of state of complex, nonlinear systems, therefore makes it an efficient tool for early fault detection.[116] Kohonen, et al.[116] have given a comprehensive review of the applications of SOM in engineering applications. In particular, they have summarized two fault identification techniques by using the quantization errors and visualization power of SOM. These two techniques have been adapted by many others to detect and identify faults for different systems.

Gonçalves, et al.[118] have utilized both techniques to detect and identify faults of electrical valves. The SOM was trained to form five feature clusters with five data sets comprising the normal condition and four fault conditions. A fault was detected when the quantization error exceeded a certain threshold. For fault identification, the dynamic behaviours of the monitored system were visualized as trajectories on SOM. The fault type was identified when the trajectory moved in one of the four fault clusters. Similar techniques for fault identification can also be found in references.[117,120-122]

Although the above research studies demonstrated the capability of SOM in dynamic monitoring and fault identification of complex systems, they are limited by the availability of data and they failed to address the potential impact of fault on the system.

In fact, the visualization power of SOM also has the capability of indicating the magnitude of fault which can be used to determine the potential impact.

One important feature of SOM is that data with high similarity are mapped closer to each other; otherwise, they are mapped further apart.[123] This provides a means of measuring the progression of fault; that is, as the fault condition deteriorates, the process system generates data with less similarity to the normal data and is mapped further away from the normal cluster. In this regard, the exceedance of fault data from normal cluster corresponds to the degree of fault and the trajectory representing the dynamic behaviour of system indicates the progression of fault.

Zadakbar, et al.[124] described a way of measuring the impact of fault using a risk-based approach. They applied Principle Component Analysis (PCA) as the dimensionality reduction technique for fault detection. The normal data was projected into a subspace determined by PCA to form a normal cluster. The cluster was considered as a standard normal distribution and its boundary was defined by mean and standard deviation of the projected data. When monitored data was projected into the same subspace, the probability of fault and exceedance of process system operation were calculated based on the mean and standard deviation. The intensity of fault at a given exceedance was determined by summing the hazard potential of each system variable. Subsequently, the severity of fault was calculated based on the intensity and exceedance. Finally, the severity of fault was combined with the probability of fault to determine the risk of fault which provided a measure of potential impact on the system. However, due to the linear nature of PCA, the sensitivity of this approach for fault detection is limited.

In this work, SOM is combined with the risk-based approach developed by Zadakbar, et al.[124]. The normal cluster on SOM is considered as a standard normal distribution. The probability, intensity and severity of fault are calculated and are combined to determine the risk of the fault. In addition, this new approach is also combined with probabilistic analysis to characterize the risk of fault into different levels. This allows a refined monitoring of the system as fault propagates. Proper safety measures and remedial actions can then efficiently be determined in correspondence to different risk levels to minimize the potential impact.

This paper is divided into the following sections: In Section 6.2, the methodology of the new approach based on SOM is explained. The verification of this new approach is then conducted in Section 6.3 on two experimental systems: a tank pressure control system in Section 6.3.1 and a flow control system in Section 6.3.2. The faults for verification are introduced as deviations into one variable of each system. The results from both systems are also discussed. Section 6.4 summarizes the major findings of the paper and conclusions are drawn.

6.2 Methodology

The overall methodology of risk-based fault detection approach is outlined by the following logical chart.

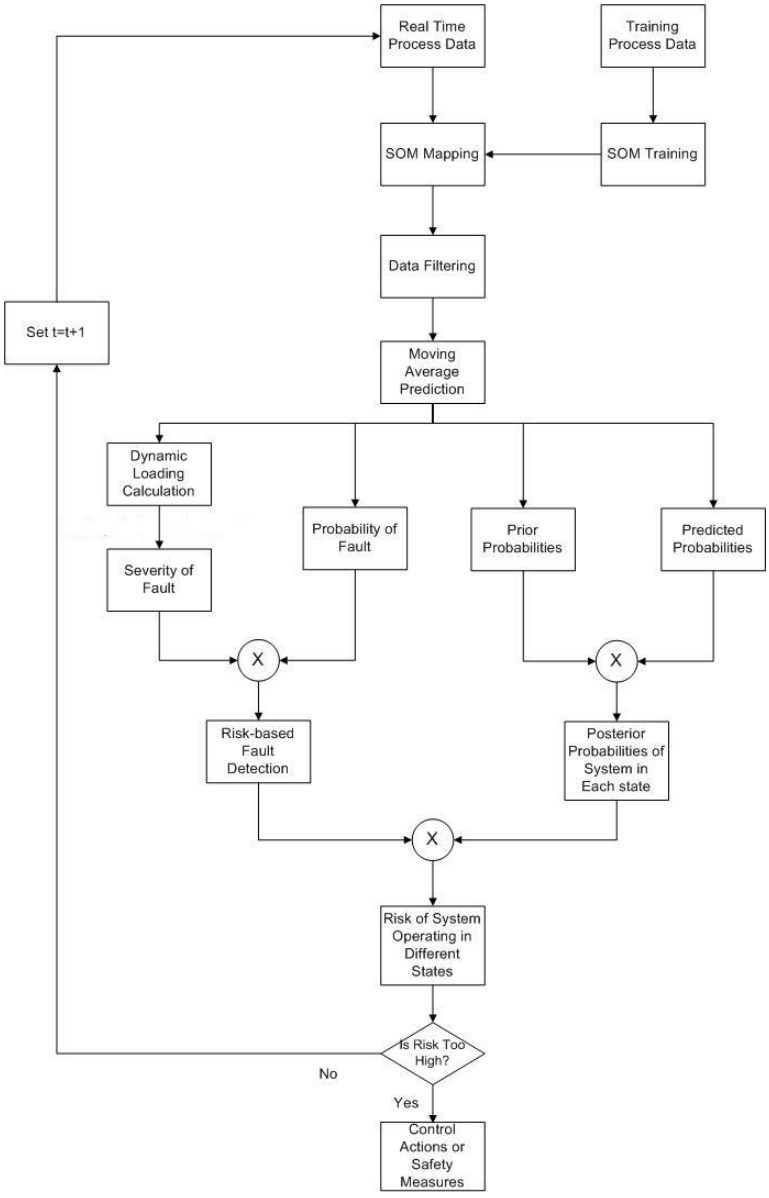

Figure 6-1: Risk-based Fault Detection Logical Flow Chart

The real-time system data is projected onto a trained SOM map to form a trajectory representing the dynamic behaviour of the system. Data filtering is then applied to the trajectory to filter out less significant variations of the system. Based on the trend of the filtered trajectory, the system behaviour is predicted five-point forward using moving average trend prediction. Subsequently, the dynamic loading, severity and probability of fault are calculated. The dynamic loading is used to identify the root cause of fault. The risk of fault is determined by combining the severity and probability of fault. Meanwhile, the operation of the system is characterized into different states through probabilistic analysis. The prior probabilities and predicted probabilities of system operating in different states are calculated. The posterior probabilities of system operating in different states are determined by updating the prior probabilities with the predicted probabilities. Finally, the posterior probabilities and the risk of fault are used to determine the risk of

system operating in different states. According to the risk level, proper remedial actions and safety measures are determined to minimize the potential impact of the fault.

6.2.1 Self-Organizing Map

The SOM was proposed by Kohonen[125] as a specific type of neural network. Its concept is originated from the functions of cerebral cortex of brain. The cerebral cortex is divided into different areas for processing signals such as sight, hearing and tactile sensation [126]. On receiving these signals, the cortex will first classify and then map them to the corresponding areas to be processed. In each area of the cortex, neurons with similar functionality are closely related, leading to fast and accurate processing of the signals. This form of classifying and mapping signals to the corresponding processing area is called topographic mapping which is also the fundamental concept of the SOM [125].

Self-organizing map is able to discover the nonlinear latent features from high dimensional data. These low-dimensional features are presented in the form of a layer of topologically ordered neurons on a 2D map. A typical two-dimensional SOM is shown in Figure 6-2.

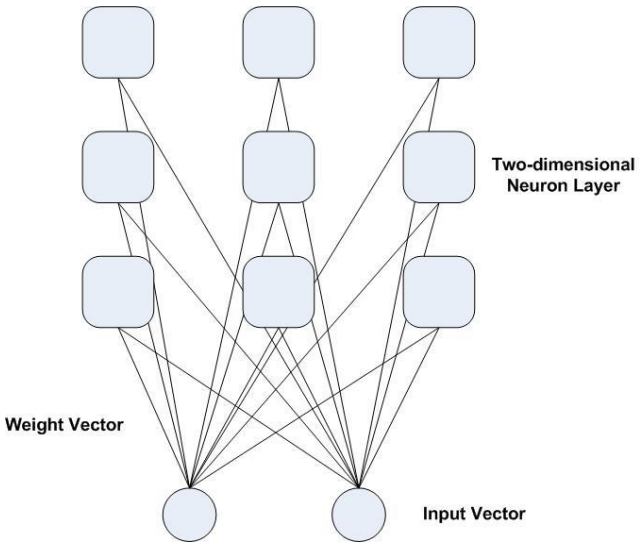

Figure 6-2: Basic SOM structures

Training of SOM mainly composes of three phases; competition, cooperation and adaption [125]. In the phase of competition, neurons first compete with each other and the neuron having the weight vector closest to the input signal vector is declared as the winner neuron or the Best Matching Unit (BMU). It is assumed the input signal vector is represented by $\mathbf{I}= [I_1, I_2, I_3, \ldots, I_n]^T$ and the weight vector is represented by $\mathbf{W}= [W_1, W_2, W_3, \ldots, W_n]^T$. Mathematically, the difference between the weight vector and the input signal vector is computed as the Euclidean Distance between them.

$$E = \| \mathbf{I} - \mathbf{W} \| = \sqrt{\sum_{i=1}^{n} (I_i - W_i)^2} \tag{6.1}$$

The neuron that has the smallest $\boldsymbol{E}$ is the BMU. Next, in the cooperation phase, the direct neighbourhood neurons of the BMU are identified. Finally, in the adaption phase, these neurons are selectively tuned to form a specific pattern on the lattice. This pattern corresponds to a specific feature of the input signal vector. The tuning function is expressed as;

$$\mathbf{W}(t+1) = \mathbf{W}(t) + \alpha(t)\theta(t)[\mathbf{I}(t) - \mathbf{W}(t)] \tag{6.2}$$

where $\alpha(t)$ is the tuning rate and $\theta(t)$ is the exponential neighbourhood function. $\alpha(t)$ decreases exponentially over iteration resulting in a more refined tuning towards the end of training process.

$$\alpha(t) = \alpha_0 e^{(-t/\lambda)} \tag{6.3}$$

where α_0 is the initial learning rate and λ is the time constant which is determined as.

$$\lambda = \frac{N}{\sigma_0} \tag{6.4}$$

where N is the total number of training samples. σ_0 is the radius of the map. It is computed as the Euclidean distance between the coordinates of the outmost neuron and the centre neuron.

$$\sigma_0 = \left\| \mathbf{T}_{outmost} - \mathbf{T}_{centre} \right\| \tag{6.5}$$

It is noted that on 2D map, the coordinates of each neuron is expressed as $\mathbf{T}_j = [t_j^1\ t_j^2]$. $\mathbf{T}_{outmost}$ denotes the coordinate of the outmost neuron while $\mathbf{T}_{centre}$ represents the coordinate of the central neuron. On the other hand, $\theta(t)$ is maximized at the BMU and decays exponentially with the distance from the BMU.

$$\theta(t) = \exp\left(\frac{\left\| \mathbf{T}_j - \mathbf{T}_{BMU} \right\|^2}{2\sigma(t)^2} \right) \tag{6.6}$$

$$\sigma(t) = \sigma_0 \exp\left(\frac{-t}{\lambda} \right) \tag{6.7}$$

where $\mathbf{T}_{BMU}$ is the coordinate of the best matching unit and $\sigma(t)$ is the radius of the neighbourhood. This means that the neurons that are farther away from the BMU are updated at a much lower rate. In addition, the neurons that are outside the radius of neighbourhood are skipped completely. As a result, the weight vectors of the BMU and its neighbours gradually become more similar to the input data samples. Conversely, the similarity between the weight vectors of the neurons farther away and the input data sample decreases over time. This type of differential tuning leads to similarity mapping of the data samples and topological ordering of the neurons. The training process stops until

the maximum number of training iterations is reached. After training, the topologically ordered neurons form a 2D pattern which corresponds to the low-dimensional latent features of the training data samples, such as the example shown in Figure 6-3. In this respect, the SOM can also be used as a data classification tool with which data samples with similar features are mapped into a single cluster.

When applying SOM to complex system, different operating states of the system are mapped as clusters on a two-dimensional map. During online monitoring, each sample data vector is compared with the weight vector of neurons within each cluster and the BMU is computed. Depending on the location of the BMU, the operating state of system is identified. By connecting the BMUs of all sample data, a trajectory is formed which shows the dynamic behaviour of the system. When the system is operating at a normal condition, the process data samples are rather similar to each other and are mapped into a single cluster. As a result, the trajectory is restricted within the cluster representing normal operation. In case of fault condition, the system is subjected to abnormal variations which lead to generation of data samples having very distinctive features. These faulty data samples are mapped in a different cluster. In addition, as the fault condition deteriorates, the data samples generated become more dissimilar and are mapped in a cluster farther away from the normal cluster. Consequently, the trajectory connecting neurons in which online process data samples are mapped deviates from the normal operating cluster and moves into the cluster representing one particular fault state.

6.2.2 Dimensionality Reduction of SOM

As a dimensionality reduction technique, SOM represents the dynamic behaviour of system as a two-dimensional trajectory. The location of each BMU of the trajectory is defined by their coordinates which consist of two variables corresponding to the two axes of SOM. In this regard, the number of variables required for system monitoring is reduced to two; therefore the dimensionality is also reduced. In essence, the dimensionality reduction of SOM is non-linear [127].

The mechanism of this non-linear dimensionality reduction can be explained from the fundamental equation

$$\mathbf{T} = \mathbf{XU} \tag{6.8}$$

Where **T** is the coordinate, **X** is the sample data vector and **U** is called the loading vector. The loading vector comprises elements that indicate the contribution of each system variable to the current operating state of the system. For every data sample collected from the system, it is transformed to the corresponding coordinates on SOM through the loading.

Eq. (6.8) is also a fundamental equation of PCA. In PCA, once the subspace for projection is determined, the loading remains stationary for all the system states. In reality, however, the contribution of each variable to different states of system does not always remain stationary. In particular, if there is a fault in the system, the contribution of the variable that is closely related to the fault will certainly increase. Therefore, the mechanism of the non-linear dimensionality reduction is interpreted as when sample data

X is varying, loading **U** is also varying along with it. During system monitoring, the coordinates of every data sample can be easily determined. The loading **U** can therefore be dynamically calculated for each data sample.

$$\mathbf{U} = \mathbf{X}^{-1}\mathbf{T} \tag{6.9}$$

By analysing the variation of the loading, it is possible to identify the variable/variables that is/are most influential to a particular fault. Subsequently, the root cause of fault is identified.

6.2.3 Determining the Dominating Axis

Depending on the nature of the fault, each axis of SOM may have different level of sensitivity to detect the fault. In practices, the axis with the highest sensitivity should be selected for fault detection. However, this is particularly difficult when the two axes are not at the centre of each cluster.

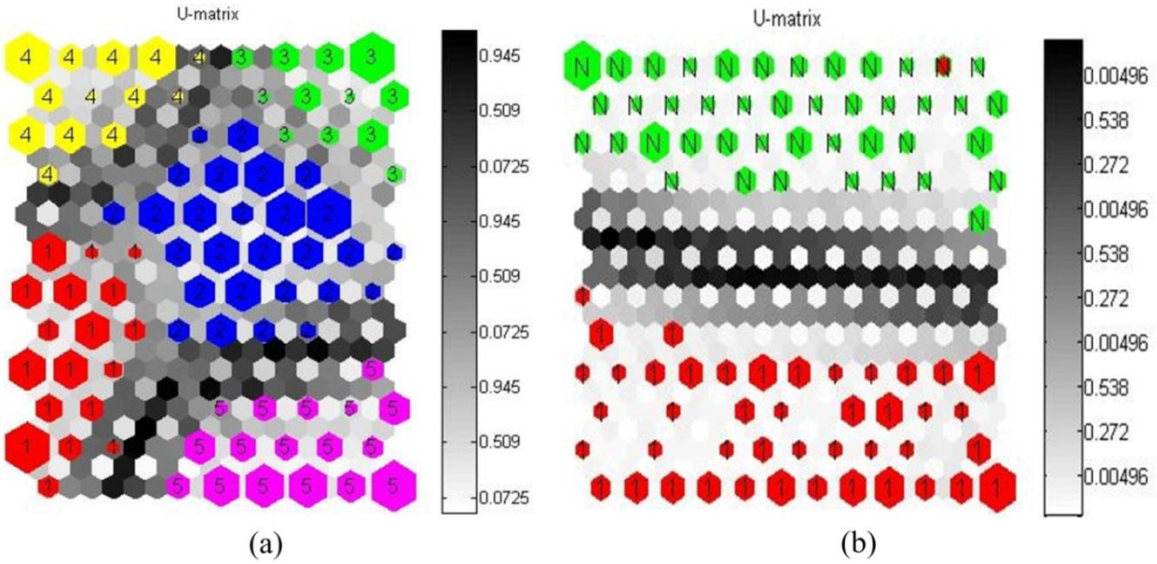

Figure 6-3: SOM Clustering

In Figure 6-3(a), five data sets with different features are mapped on SOM. This has resulted in five different clusters. In system monitoring, these five data sets are collected from five types of operating conditions of the system (Green represents the normal condition and others represent faulty conditions). Gonçalves, et al.[118] has applied this method for fault diagnosis. This map is only limited to identify four types of fault.

Since there are no individual axes for each cluster, when a data sample is mapped in the top-right corner of cluster 2, using X axis as the dominating axis, the system is identified as having condition 2, 3, 5 at the same time which is physically not realistic. Likewise, when Y axis is used, the system is identified as having condition 2, 3 and 4 at the same time. To address this issue, the number of clusters has to be reduced and SOM is to be trained in a manner that the cluster representing the normal operating condition occupies the span of one axis. This axis then becomes least significant and the other axis is forced to capture all the variations of system.

In Figure 6-3(b), only two sets data are collected: one being collected from normal operating state and the other being collected from a random faulty operation state. The number of clusters is reduced to two. The span of X axis is completely occupied. The Y axis is forced to be the dominating axis. As outlined in Section 6.2.1, when the process system operation is within normal regime, the online data samples generated are very similar to the normal training data; the trajectory stays within the normal green cluster. After a fault condition is injected into the system, the trajectory starts to diverge due to the generation and mapping of the very dissimilar online faulty data. The Fault condition is detected when the dynamic trajectory deviates out of the green cluster. Root cause identification can then be conducted by analysing the behaviour of the loading variation of each system variable. The types of faults can be detected is only limited by the number of possible combination of the system variables.

6.2.4 Risk-based Fault Detection

When applying SOM for system monitoring, there are two major fault detection techniques. The first focuses on the visualization power of SOM; a fault is detected when the mapped data deviates from a defined cluster or path on SOM [116,122,128,129]. The second method focuses on the quantization error of the SOM [116,118]. However, the above methods assume equal hazard of the faults detected and do not consider their potential impact on the system. This results in no differentiation between faults generated by trivial changes and the catastrophic failures of the system.

A risk-based fault detection technique is developed for SOM based on the work of [124]. Risk is a quantitative measurement of potential loss caused by the fault if proper remedial action is not taken promptly. Risk-based fault detection provides a robust differentiation between faults with low hazard and high hazard to the system. Risk depends on two factors: the probability of a fault occurring and the severity of the consequence of a fault. Bao, et al.[130] proposed a quantitative analysis of risk using the following equation.

$$Risk = P \times S \tag{6.10}$$

P is the probability of fault occurring and S is the severity of the consequence of the fault. The probability of a fault occurring increases as the system deviates further from the normal operating condition. This behaviour can be captured using the probability density function of a cumulative normal distribution.

$$P = F(t \mid \mu, \sigma) = \begin{cases} \varphi\left(\dfrac{t-\mu}{\sigma}\right) & t \geq \mu \\ 1-\varphi\left(\dfrac{t-\mu}{\sigma}\right) & t < \mu \end{cases} \tag{6.11}$$

$$\varphi\left(\frac{t-\mu}{\sigma}\right) = \int_{-\infty}^{t} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(t-\mu)^2}{2\sigma^2}} dt \tag{6.12}$$

In the case of SOM, the dynamic behaviour of system is mapped as a trajectory on SOM. t is the coordinates of the BMUs on the trajectory corresponding to the dominating

axis. The normal operating cluster is considered as a normal distribution with mean μ and standard deviation σ. The upper and lower boundaries of the cluster are defined as μ+3σ and μ-3σ respectively which comprise 99.73% of the coordinates within the normal cluster. Equation (6.11) and (6.12) are modified to calculate the probability of the trajectory exceeding the upper and lower boundaries; that is, system exceeding the normal operating cluster leading to a fault.

$$P_i = F(t_i \mid \mu, \sigma) = \begin{cases} \varphi\left(\dfrac{t_i - (\mu + 3\sigma)}{\sigma}\right) & t_i \geq \mu \\ 1 - \varphi\left(\dfrac{t_i - (\mu - 3\sigma)}{\sigma}\right) & t_i < \mu \end{cases} \tag{6.13}$$

$$\varphi\left(\frac{t_i - (\mu \pm 3\sigma)}{\sigma}\right) = \int_{-\infty}^{t_i} \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{(t_i - (\mu \pm 3\sigma))^2}{2\sigma^2}} dt \tag{6.14}$$

where $\in [1, n]$, n is the number of data samples. The severity of a fault depends on the exceedance of process operation from normal and the intensity of the undesired condition.[130] In this work, the exceedance of process operation is calculated as how far the dynamic trajectory on the 2D SOM has deviated from the normal cluster. As outlined in Sections 6.2.1 and 6.2.3, the deviation of trajectory is due to the generation and mapping of very dissimilar faulty data. As the fault condition deteriorates, the process operation deviates further away from its set-point condition. In this case, the exceedance is calculated using the modified version of equation proposed by Bao, et al.[130]. This modification is necessary as a two-dimensional SOM only allows two degrees of freedom.

$$E_i = \begin{cases} 2^{\frac{t_i - (\mu + 3\sigma)}{t_i - \mu}} & t_i \geq \mu \\ 2^{\frac{t_i - (\mu - 3\sigma)}{t_i - \mu}} & t_i < \mu \end{cases} \tag{6.15}$$

In fact, in the above equation, the exponential term is a measure of how far the trajectory exceeds the boundary of the cluster, which is defined by three standard-deviations from the mean. On the other hand, the intensity provides a relative measure of the hazard potential of a fault at given exceedance. It takes into account the hazard potential of each process variable and their instant contribution to an undesired condition. An intensity coefficient is assigned to each system variable based on their relative hazard potential; a system variable with relatively high hazard potential is assigned with a larger intensity coefficient. This mechanism is demonstrated in the two case studies presented in the subsequent section. As mentioned in Section 6.2.2, the contribution of each system variable to a particular fault is indicated by their loadings. Therefore, the intensity of a fault is calculated using the following equation.

$$In_i = \sum_{j=1}^{m} a_j u_{ij} \tag{6.16}$$

where a_j is the intensity coefficient assigned to process variable j, u_{ij} is the dynamic loading of process variable j at sampling instant i, m is the total number of monitored process variables. It is easily seen that the instant intensity of fault is a weighted summation of the instant intensity contributed by each process variable. Subsequently, the severity of a fault is calculated through Equation(6.17).

$$S_i = E_i \times In_i \tag{6.17}$$

Finally, the risk of a fault is determined using Equation(6.18).

$$Risk_i = P_i \times S_i \tag{6.18}$$

It has to be noted that u_{ij} is varying from sample to sample. It is dynamically calculated using Equation(6.9). Equation (6.9) is a matrix operation which requires the size of matrices **U**, **c** and **X** to be compatible with each other; if **U** is a 1 by m matrix, **X** has to be an m by m square matrix and **c** has to be a 1 by m matrix. In SOM, data samples with similar variations are mapped in one neuron. Therefore, to solve U, the number of data samples that are mapped in a neuron has to be equal to the number of system variables. A neuron in which a large number of data samples are mapped represents significant variations of the system. Conversely, neurons in which a smaller number of data samples are mapped represents less significant variations of the system. For this reason, when solving U, these neurons are neglected. In essence, this is a data filtering technique based on SOM that filters out the less significant information from the system.

This data filtering technique is demonstrated in Figure 6-4(a) and (b). Two different data sets which are collected from normal and fault condition of a system are mapped into two clusters on SOM. A new data set is generated that contains features of both conditions. This new data set is then mapped on the same SOM. Figure 6-4(a) shows the trajectory of the new data set without filtering. Figure 6-4(b) shows the trajectory of the new data set with filtering. As compared to Figure 6-4(a), the trajectory in Figure 6-4(b) clearly demonstrates the progression of system state from normal to fault condition.

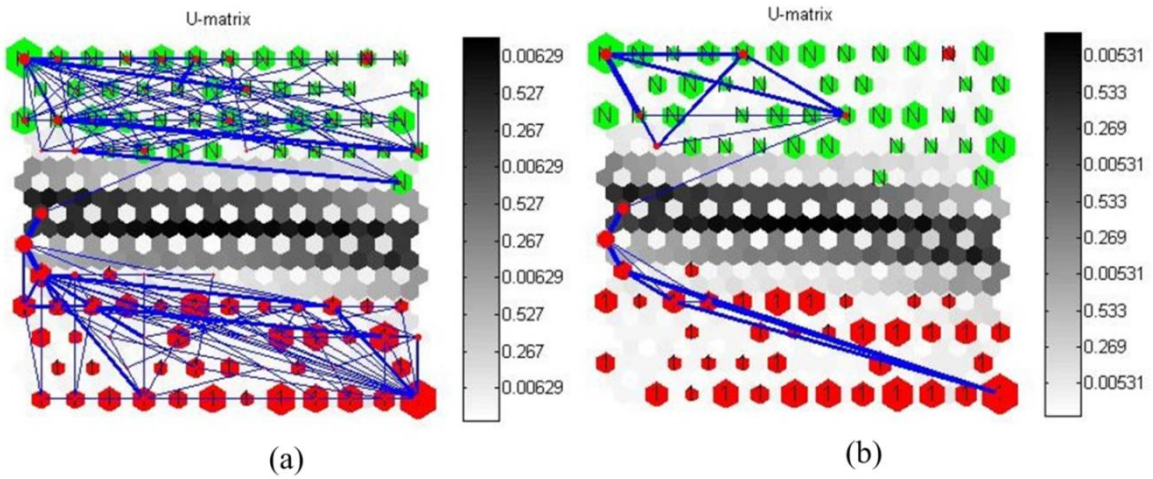

Figure 6-4: SOM Trajectory with Filtering

In next step, the sensitivity of this approach is further improved by integration with prediction capability. Instead of providing an instant measure of the risk, the new approach predicts the risk of fault few instants in advance and gives an earlier indication of fault. Moving average trend prediction is applied on the trajectory of SOM to predict the coordinate five-point forward. The risk of fault is then calculated based on the predicted coordinate using Equation(6.13) through Equation(6.18).

6.2.5 Probabilistic Analysis

The probabilistic analysis provides the probabilities of system operating in different states after a fault has occurred. This analysis is conducted using a technique known as Bayesian Updating. When Bayesian Updating is used for system monitoring, the probabilities of system operating into different states are calculated, e.g. safe operation, degraded performance, and accident. This provides the advantage in refined monitoring of the system and efficient determination of safety measures and remedial actions to reduce the potential of an accident.

With regards to safety measures and remedial actions, it is possible to integrate the developed technique in the framework of Event Tree Analysis which allows the analysis of consequences of failure of different safety barriers [131]. This type of analysis also helps to assess the effectiveness of various safety measures (barriers). However, in the present study, the focus of this work is the refined monitoring of the process system operation; it is assumed that no safety measures and remedial actions are implemented. In this regard, the ETA demonstrated is only a framework for future research.

The operation of the system is characterised into five states reflecting different risk levels. These states are listed as following.

1. Normal
2. Control
3. Warning
4. Shutdown

5. Accident

The normal state indicates the system is operating in a normal operating condition with minimum risk. Control state indicates the risk of system has exceeded the normal operating condition. With proper control action, the risk level can be brought back to normal. If the control action fails, a higher level of risk could be reached which will activate the alarm of the safety system. If the safety system fails to confine the risk, the risk level could further increase and eventually leads to the shutdown of the system. The worst case scenario is the occurring of an accident which is the consequence of failure of the shutdown system. This characterization allows a more refined monitoring of the system. Depending on the risk level, proper control actions and safety measures can be taken in advance to prevent the shutdown of the system and the occurrence of an accident.

The region of each state associated with different risk level is defined on SOM using the mean μ and the standard deviation σ of the normal operating cluster. In section 6.2.4, the normal state is defined between [μ-3σ, μ+3σ] which contain 99.73% of the coordinates within the normal cluster. The control state represents coordinates that exceed the normal state and is defined between [μ+3σ, μ+4σ]. In analogy, the warning state is defined between [[μ+4σ, μ+5σ] and the shutdown state is defined between [μ+5σ, μ+6σ]. The accident state is beyond the shutdown state which covers from μ+7σ to the border of the SOM. The exceedance of coordinates is increased from one state to the next state.

$$\begin{cases} t_i \in [\mu-3\sigma, \mu+4\sigma], \text{ Normal, state 1} \\ t_i \in [\mu+4\sigma, \mu+5\sigma], \text{ Control, state 2} \\ t_i \in [\mu+5\sigma, \mu+6\sigma], \text{ Warning, state 3} \\ t_i \in [\mu+6\sigma, \mu+7\sigma], \text{ Shutdown, state 4} \\ t_i \in [\mu+7\sigma, t_{\max}], \text{ Accident, state 5} \end{cases} \tag{6.19}$$

where t_{max} is the maximum coordinate on the SOM. The prior probability of the system operating in each state is calculated based on the historical data (prior knowledge) of the system. The system is first monitored for a period of time. The monitored data is then mapped on SOM to determine the coordinates and the dynamic loadings. The number of coordinates appear in each state is also determined. The prior probability of each state is then calculated using the following equation.

$$P(K) = \frac{N_K}{N_T} \tag{6.20}$$

where $K \in \{1,2,...5\}$ is the region number, N_K is the number of coordinates in region K and N_T is the total number of coordinates. The next step is prediction. Moving average trend prediction is applied on the trajectory to predict the coordinates five-point forward. The probability of system operating in each state is calculated specifically on the five predicted coordinates using the following equation.

$$P(K_p) = \frac{N_{Kp}}{N_{Tp}} \tag{6.21}$$

where $K_p \in \{1,2,...5\}$ is the region number for predicted coordinates, N_{Kp} is the number of coordinates in region K_p and N_{Tp} is the total number of predicted coordinates which is equal to 5. The predicted coordinates are dependent on the historical data. Logically, the predicted probability $P(K_p)$ is also dependent on the probability $P(K)$, that is, $P(K_p)$ can be interpreted as a conditional probability dependent on $P(K)$. Subsequently, the prior probability $P(K)$ is updated using the predicted probability $P(K_p)$ via the Bayesian updating. The Bayesian updating rule is expressed as follow.

$$P(A \mid B) = \frac{P(A) \mid P(B \mid A)}{\sum_A P(A) \mid P(B \mid A)} = \frac{P(A \cap B)}{P(B)} \tag{6.22}$$

where $P(A)$ is known as the prior probability, $P(B|A)$ is known as the conditional probability (or observation in some case) and $P(B)$ is the normalization probability. $P(B)$ is also the probability of event B across all possible scenarios of A. $P(B|A)$ is the posterior probability. In this study, the prior probability is $P(K)$ and the conditional probability is $P(K_p)$. On the other hand, the normalization probability $P(K_t)$ (equivalent to $P(B)$) is the overall probability of the system operating in each state for both the historical and predicted cases (scenarios). In this respect, the posterior probability of the process system operating in each state can be updated using the following equations.

$$\begin{aligned} P(K_u) &= \frac{P(K) \times P(K_p)}{P(K_t)} \\ P(K_t) &= \frac{N_K + N_{Kp}}{N_T + N_{Tp}} \end{aligned} \tag{6.23}$$

where:

$K_u \in \{1,2,...,5\}$, region number for updated probability;

$K_t \in \{1,2,...,5\}$, region number for the normalization probability;

$N_K + N_{Kp}$ is the total number of coordinates in region K for both the historical and predicted scenarios;

$N_T + N_{Tp}$ is the total number of coordinates in all regions for both the historical and predicted scenarios;

$P(K_u) =$ the posterior probability;

$P(K_t) =$ the normalization probability.

The major advantage of updating the probabilities using the predicted coordinates is to provide some lead time for both fault detection and implementation of safety actions. Next, the risk of the system operating in each state can be calculated using Eq. (6.15) through Eq.(6.18). In addition to the intensity factor given to each system variable, an intensity factor is given to each system operating state according to the potential impact of each state if no actions are taken to minimize the risk. The intensity factor for each state is listed below.

$$A = \begin{bmatrix} A_c = 1 \\ A_w = 2 \\ A_s = 3 \\ A_a = 4 \end{bmatrix} \tag{6.24}$$

The control state is an intermediate state between the normal state and warning state. It acts as a buffer zone which provides reaction time for control action to bring the system back to the normal state. The control state has minimal impact on the safe operation of system. On the other hand, the accident state of a system may cause material losses, damage of equipment and surrounding environment. In worst case, it may put the safety of the personnel in peril. Therefore, the control state is given the lowest intensity factor while the accident state is given the highest intensity factor.

Finally, the risk of system operating in each state is calculated as;

$$\begin{aligned} Risk_i(C) &= P_i(K_u = 2) \times S_i \times A_c \\ Risk_i(W) &= P_i(K_u = 3) \times S_i \times A_w \\ Risk_i(S) &= P_i(K_u = 4) \times S_i \times A_s \\ Risk_i(A) &= P_i(K_u = 5) \times S_i \times A_a \end{aligned} \tag{6.25}$$

6.2.6 Event Tree Analysis for future development

Upon obtaining risk of the process system operation, various safety measures and remedial actions can be brought into place to minimize the economic loss and risk of catastrophic failure. The effectiveness of these measures can be assessed by conducting Event Tree Analysis. The Event Tree Diagram for this type of analysis is presented in Figure 6-5.

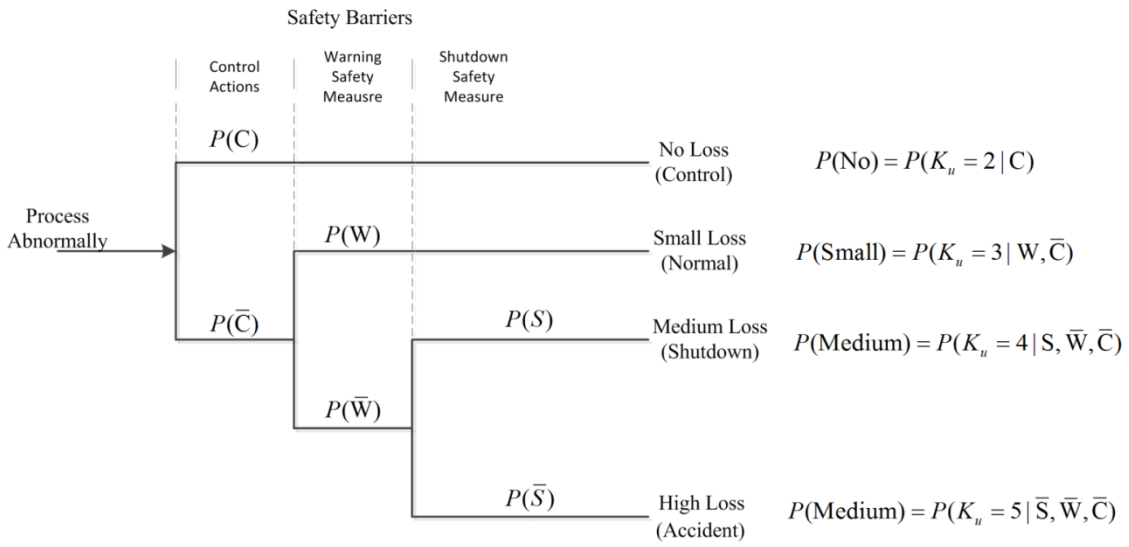

Figure 6-5 Event Tree Diagram for Consequence Analysis and Assessment of Safety Measures.

where:

$P(C)$ = probability of the control actions function properly;

$P(W)$ = probability of the warning safety measure working;

$P(S)$ = probability of the shutdown safety measure working;

$P(\bullet)$ = the complementary probabilities of the above probabilities;

$P(\text{No})$ = posterior probability of system in control region given the control action is working;

$P(\text{Small})$ = posterior probability of system in warning region given the warning safety measure is working;

$P(\text{Medium})$ = posterior probability of system in shutdown region given the shutdown safety measure is working;

$P(\text{High})$ = posterior probability of system in accident region given all safety measures have failed.

The initial event (process abnormally) could lead to four consequences. For the first two consequences, the process system operation can be brought back to normal with zero or small losses. The probability of each of the consequences can be calculated using the same Bayesian Updating method while taking into consideration the working or failure of the corresponding safety barrier. These posterior probabilities are then normalized to sum up to 1 to comply with the properties of the event tree diagram. In the case studies, to simply demonstrate the effectiveness of the proposed risk-based fault detection technique, no control actions or safety measures are taken to minimize the process losses. The framework presented in Figure 6-5 forms the basis for next stage development of this research work.

6.3 Case Study

The performance of the proposed approach is verified using experimental setups of two process systems. The first system is a tank pressure control system. The second system is a simplified flow control system. Both systems comprise three variables.

6.3.1 Pressure control

A simplified schematic drawing of the tank pressure control experimental system is shown in Figure 6-6.

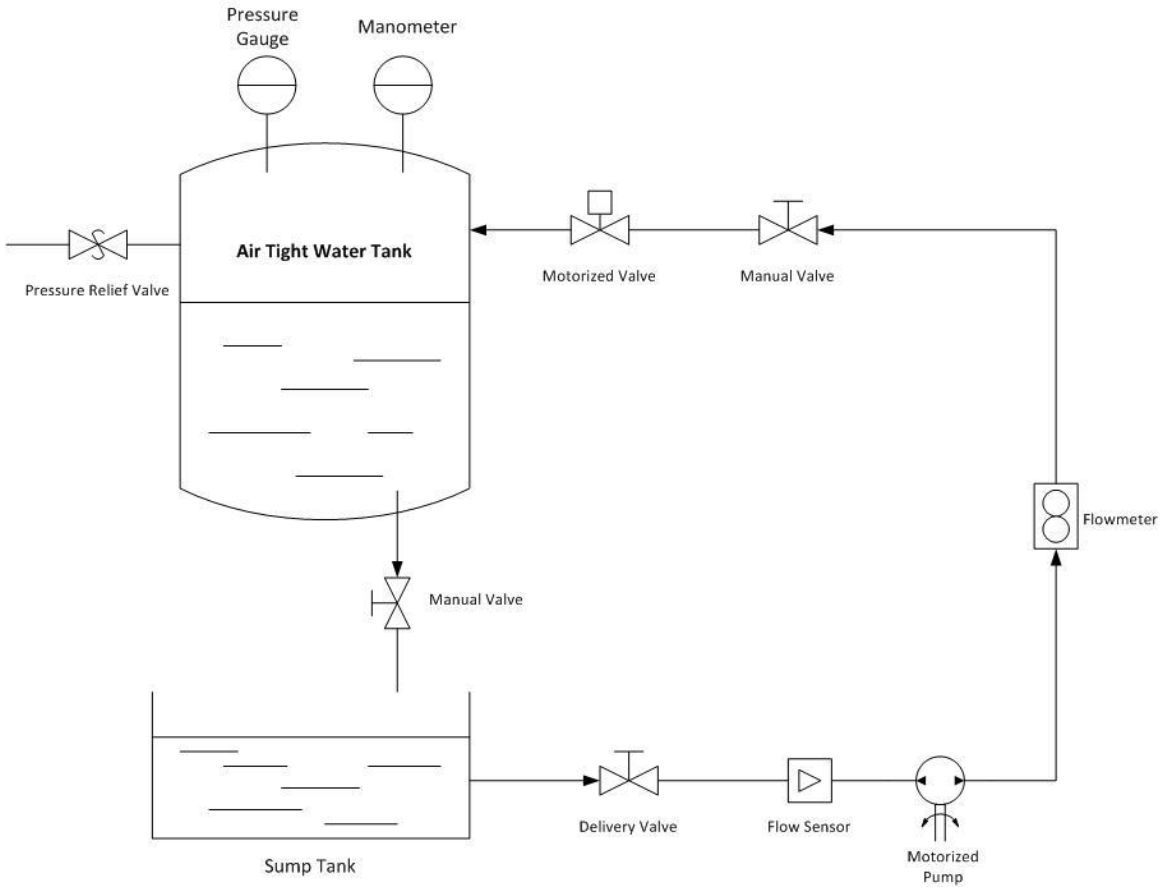

Figure 6-6: Tank Pressure Control System

The aim of the control system is to maintain the air pressure in the air tight water tank. This is achieved by varying the inlet flow rate. The inlet flow rate is proportional to the rotational speed of the motorised pump which is controlled by voltage signal from a computer controller. The three monitored variables are flow rate (Q_{in}), level (L), and pressure (P).

$$X = \begin{bmatrix} Q_{in} \\ L \\ P \end{bmatrix} \tag{6.26}$$

Monitored data from these three variables are collected from the flow sensor, level sensor, and pressure gauge respectively. In this case study, the dry down condition of the sump tank is considered as major fault. This condition causes decrease and fluctuation in flow rate which make it difficult to maintain the pressure in the tank. Figure 6-7 shows the behaviour of each system variable when the fault is introduced.

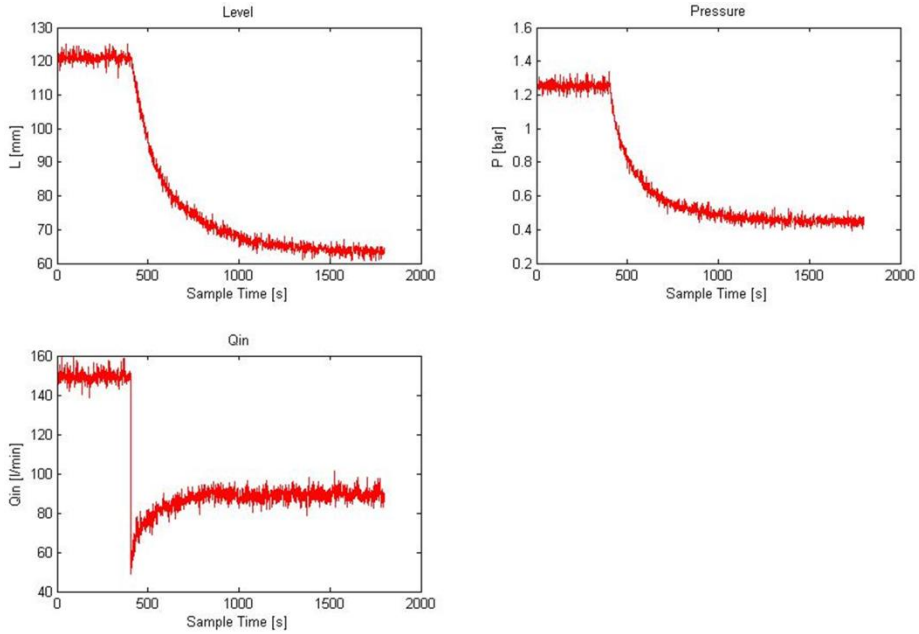

Figure 6-7: System Responses of Fault

The dry down condition is introduced at 400 seconds and causes sudden drop in flow rate. After the sudden drop, flow rate gradually increases to a steady state. In the meantime, the level and pressure drop drastically when the fault is introduced. The slop of this drop reduces as the flow rate gradually increases. Finally, both level and pressure reach a steady state.

The SOM is trained with both normal operating data and a random fault data to form two clusters. The normal operating data is generated by operating the tank system in a fault free condition for a period of time. The random fault data is generated by introducing random deviations in the obtained normal operating data. Monitored data from the dry down condition is then mapped on the trained SOM. The BMU for each data sample is computed. A trajectory is formed by connecting all the BMUs as seen in Figure 6-8 (a).

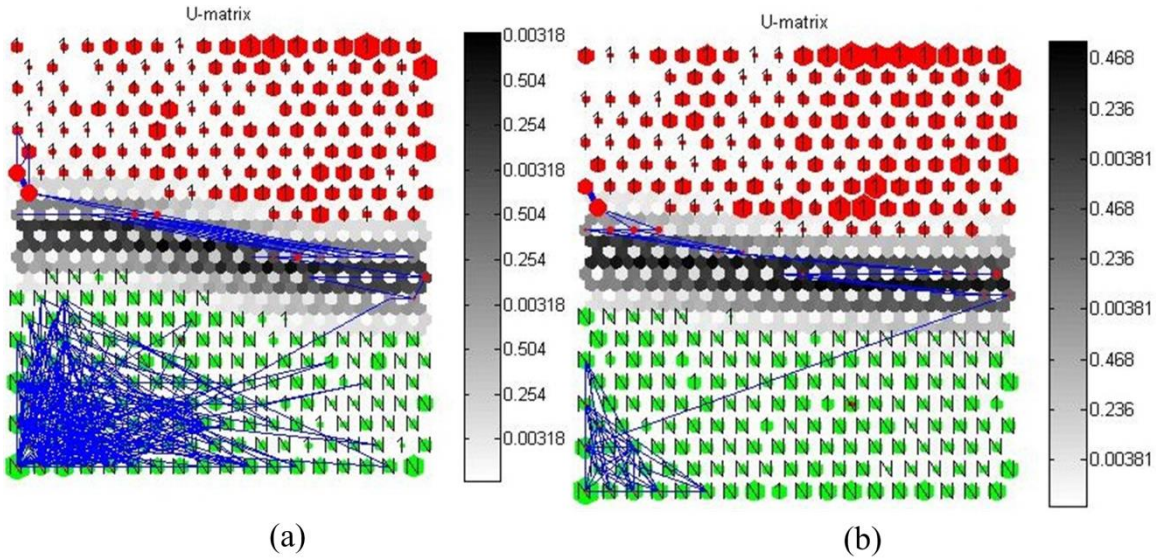

Figure 6-8: Tank Pressure Control SOM Trajectories

This trajectory represents BMUs in which all data samples are mapped. Data samples with high similarity are mapped in one BMU. Those BMUs with more data samples mapped represent significant variation of the system. Conversely, those BMUs with less data samples mapped represents less significant variations and are disregarded. In this case, it is considered BMUs with less than 10 data samples mapped are less significant and are filtered out from the SOM. The filtered trajectory is shown in Figure 6-8 (a).

As compared to Figure 6-8 (a), Figure 6-8 (b) clearly demonstrates the progression of system state from normal to fault condition. Subsequently, the loading vector corresponding to each remaining BMU is calculated using Equation(6.9). Figure 6-9 shows the variation of loading of each variable.

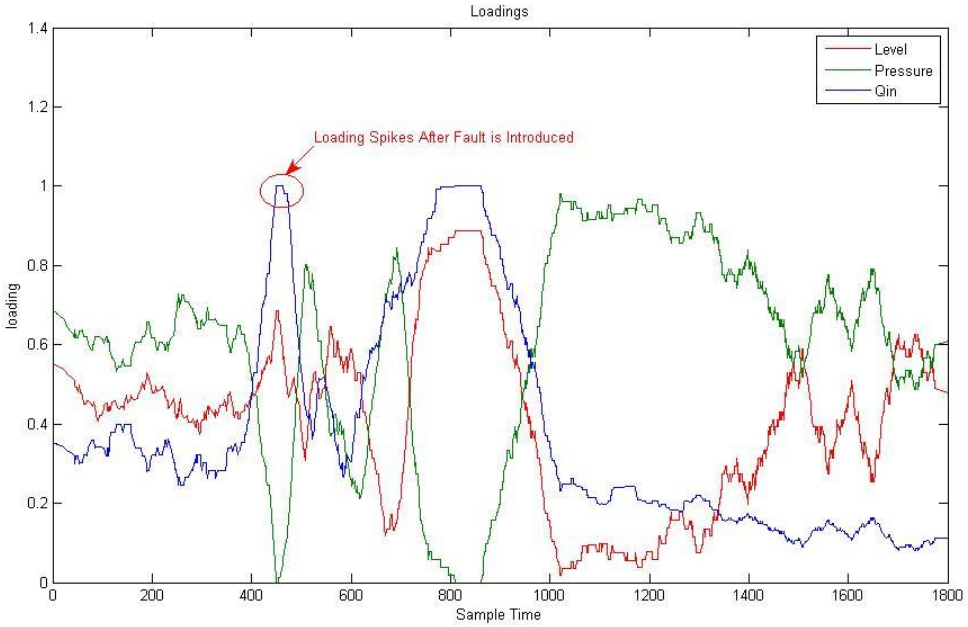

Figure 6-9: Dynamic Loading of Each Variable

At 400 seconds, the loading of flow rate increases drastically indicating increase in contribution to the fault condition. After 1400 seconds, the system reaches a steady state. The loading of each variable restores to the original condition as before 400 seconds. The dynamic behaviours of the loading not only give an early indication of fault but also facilitate the identification of the root cause. In this case, the root cause of this fault is identified to be directly related to the flow rate.

Next, the mean and standard deviation of the coordinates within the normal cluster are determined. The probability of the trajectory exceeding the normal cluster and the exceedance are calculated based on the predicted coordinates using Equations (6.13) and(6.14). To determine the intensity of the fault, an intensity factor is assigned to each system variable. In this case, the following intensity factors are assigned to the monitored variables.

$$a = \begin{bmatrix} a_{Q_{in}} = 2 \\ a_L = 1 \\ a_P = 3 \end{bmatrix} \tag{6.27}$$

Flow rate is given the highest intensity factor as large fluctuation in flow rate can cause damage to the flow meter and a sharp increase in pressure. Pressure is given the second highest intensity factor; a high pressure could cause damage to the pressure gauge, however, in this case, it is regulated by a pressure relief valve. Level is given the lowest intensity factor as it possesses minimum hazard potential to the system. The intensity of the fault is calculated using Equation(6.16). The severity of the fault is calculated using Equation(6.17). Finally, the risk of the system is determined by Equation (6.18) and is plotted in Figure 6-10.

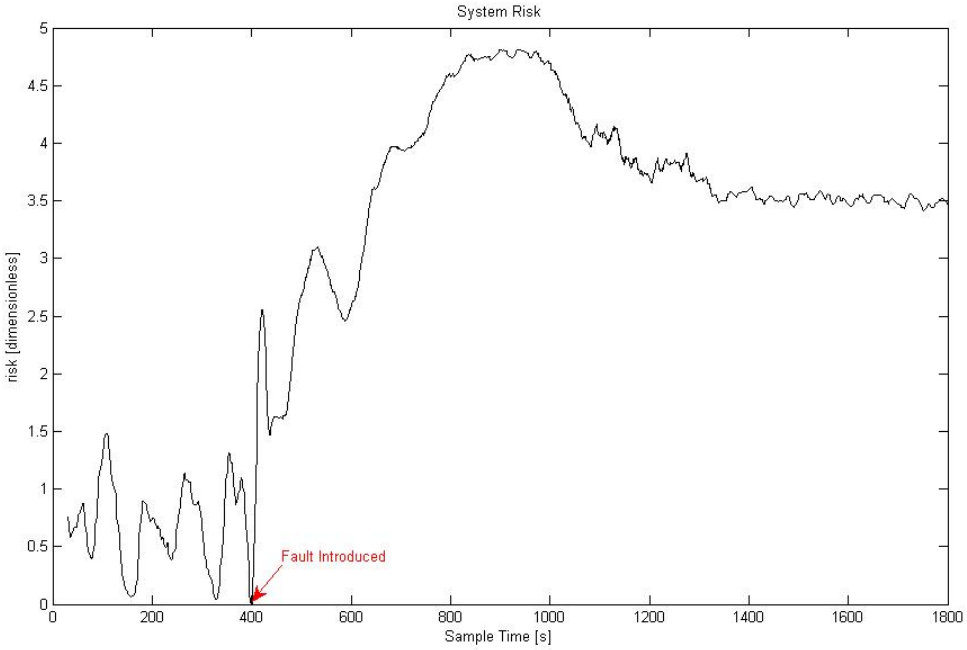

Figure 6-10: System Risk Based on Predicted Coordinates

The risk of fault gives a measure of the potential impact of fault on the system. As shown in Figure 6-10, the risk of fault spikes at 400 seconds which is the exact moment of the fault occurring. The risk increases as the fault progresses indicating an increasing potential impact. This new approach shows a very high sensitivity of change of system state. It is able to detect and assess the potential impact of fault at its early stage.

In the next stage, the risk of fault is broken down into different levels to enable a refined monitoring of the system and an efficient determination of remedial actions and safety measures. The risk-based monitoring of the tank pressure control system is shown in Figure 6-11.

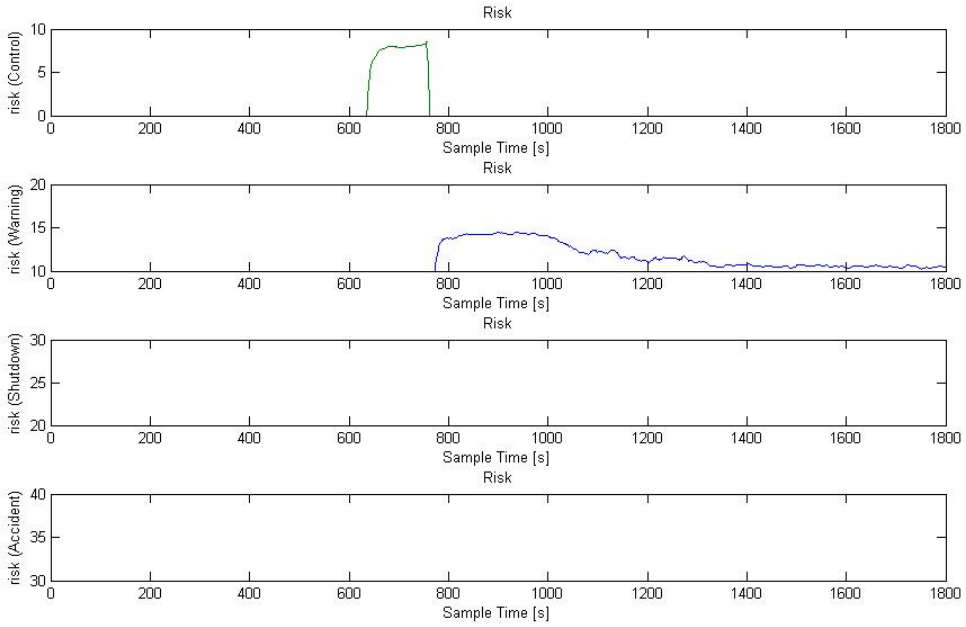

Figure 6-11: Risk-based Monitoring of the Pressure Control System

The system is operating within the range of normal state until approximately 630 seconds. After 630 seconds, the system deviates out of the normal state and enters the control state. The risk of fault increases up to 9 and stays steady until 820 seconds. By taking proper control action, risk of the fault could be brought back to normal. In this case, no control actions are taken, the risk of fault continues to increase, eventually leading to the system operating in the warning state. After 1400 seconds, the operation of the system becomes stabilized. The risk of fault becomes steady and stops increasing further into the shutdown state.

The proposed approach has demonstrated high sensitivity of change of system's state in this case study. The fault is detected at the moment of occurring. In the meantime, the root cause of the fault is identified to be directly related to the flow rate at the moment of fault occurring. The breakdown of risk into different levels allows refined monitoring of system operating states. This provides the advantage in early deployment of remedial actions or safety measures to minimize the risk of an accident.

6.3.2 Flow control

In the second case study, the proposed approach is applied to a simplified flow control experimental system. The schematic drawing of the flow control system is shown in Figure 6-12.

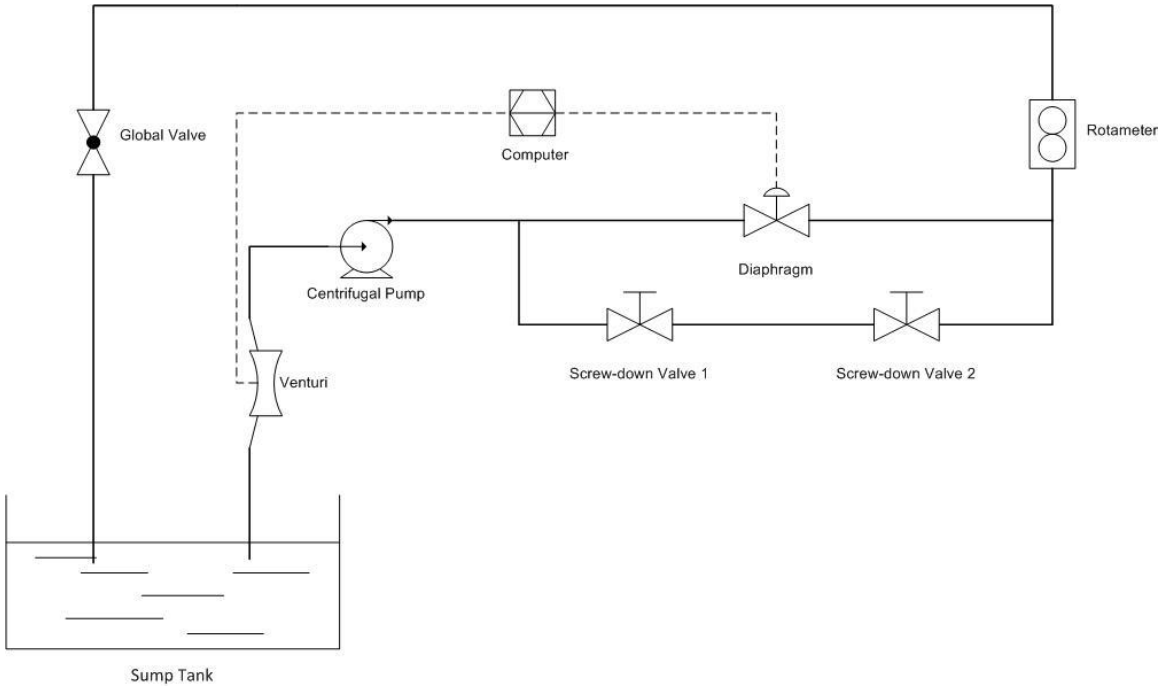

Figure 6-12: Flow Control System

Three variables including flow rate (Q), diaphragm valve opening (L) and control signal (I) are monitored to identify the operating states of the system.

$$X = \begin{bmatrix} Q \\ L \\ I \end{bmatrix} \tag{6.28}$$

Water is drawn from a sump tank by a centrifugal pump and is supplied at a constant flow rate to the system. Water flow then passes through two pipe branches. In the first branch, water flow rate is regulated by a diaphragm valve. A computer controller sends a control signal to control the pressure supply to the diaphragm valve which in turn controls the opening of the valve. In the second branch, water flow rate is regulated by two screw-down valves. In normal condition, both screw-down valves remain fully open. In fault condition, one of the screw-down valves is closed to simulate the blockage condition in the second pipe branch. Due to the blockage, flow rate in the first branch is forced to increase and the diaphragm valve is forced to operate at a higher percentage of opening. When the fault condition is introduced, the behaviour of each monitored variable is shown in Figure 6-13.

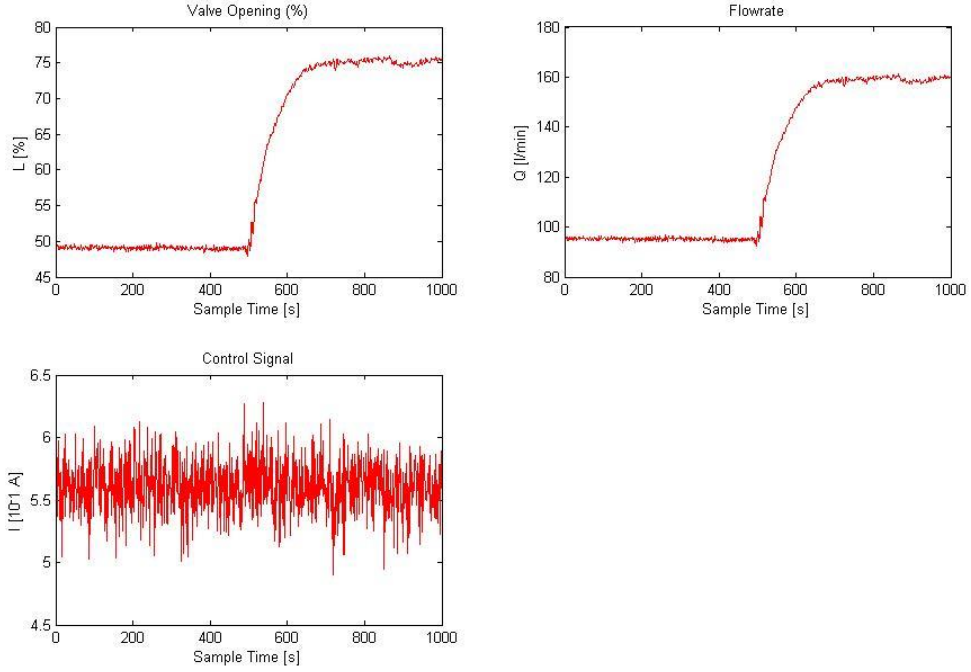

Figure 6-13: System Responses of Fault

The blockage condition is introduced at 500 seconds. This fault condition causes the sharp rise in the flow rate in the first pipe branch and the diaphragm valve opening. The control signal from the computer controller is not affected by the fault. The SOM is trained in a similar manner as case study one with data collected from normal condition and a random fault data. Monitored data from the blockage condition is then mapped on the trained SOM. The BMU representing significant variations of system are determined and connected to form a trajectory on SOM (Figure 6-14).

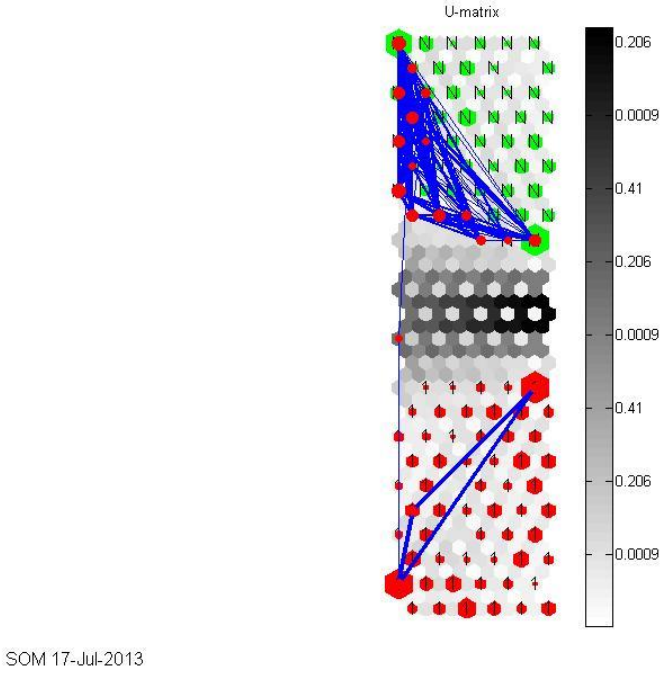

Figure 6-14: Flow Control SOM Trajectory (with filtering)

The dynamic loading of each variable to the blockage condition is shown in Figure 6-15. At 400 seconds, the dynamic loading of the flow rate increases above the loading of the valve opening which indicates high contribution to the fault condition. The root cause of this fault is identified to be directly related to the flow rate. This is true as the direct effect of the blockage condition is the decrease in flow rate. The loading of control signal remains constant throughout the process as it does not contribute to the fault condition.

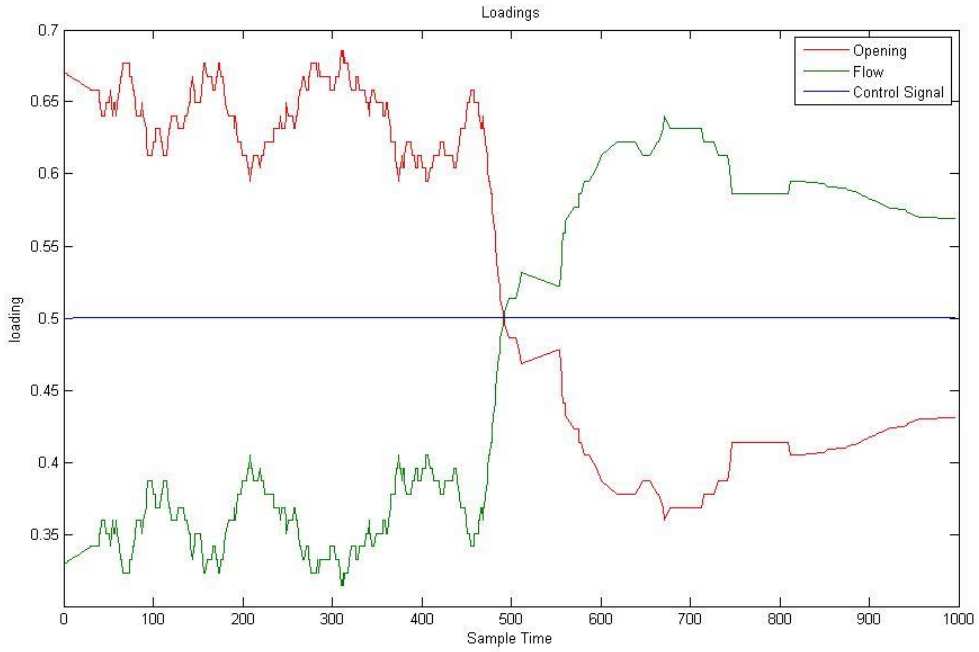

Figure 6-15: Dynamic Loading of Each Variable

Similar to the first case study, the probability of the trajectory exceeding the normal cluster and the exceedance are calculated based on the predicted coordinates using Equation (6.13) and Equation(6.14). The following intensity factors are assigned to the system variables to determine the intensity of fault.

$$a = \begin{bmatrix} a_L = 1 \\ a_Q = 2 \\ a_I = 1 \end{bmatrix} \tag{6.29}$$

In this case, flow is given the highest intensity factor as excessive flow may cause damage to the rotameter and venturi. Control signal is sent from the computer controller to control the valve opening to restrict the water flow within a safe range. These two variables are coupled and possess minimum hazard potential to the system. They are both assigned with the lowest intensity factor. Subsequently, the severity of fault is calculated using Equation(6.17). Finally, the risk of fault is the combination of the probability and the severity of fault and is calculated by Equation(6.18). The risk profile of the system is shown in Figure 6-16. The fault is detected at 500 seconds which is the moment the fault is introduced. This demonstrates the high sensitivity of the proposed approach in fault detection.

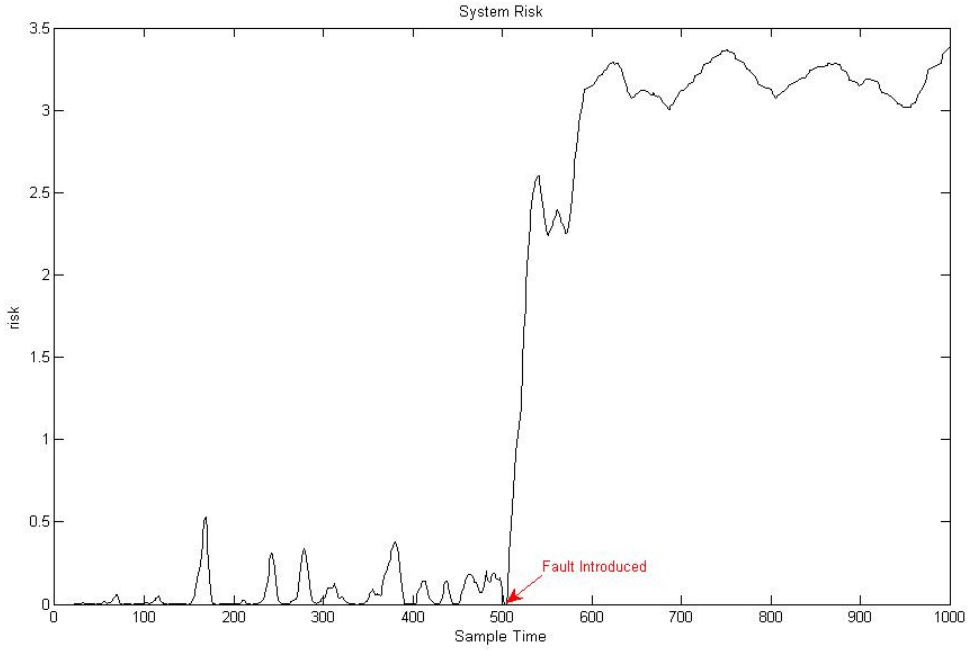

Figure 6-16: System Risk of the Flow Control System

This risk is then further broken down into different levels to enable a refined monitoring of system which leads to efficient interventions. The risk breakdown of the flow control experimental system is shown in Figure 6-17.

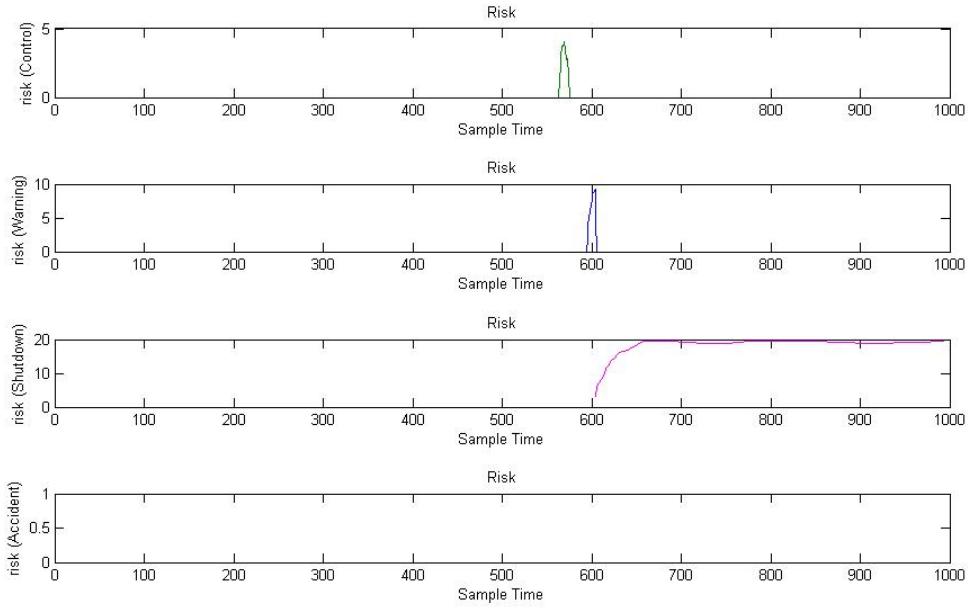

Figure 6-17: System Monitoring with SOM

After the fault is introduced, the operation of system deviates out of normal state at approximately 570 seconds. Without any control actions being taken, the risk of fault

increases in warning state. After 600 seconds, the risk of fault increases further into the shutdown state. The risk of fault becomes stabilized when the system reaches a steady state after 700 seconds.

The results of the second case study demonstrate high sensitivity of the proposed approach to the change of system state. Fault is detected and the root cause is identified at the moment of occurring. In combination of probabilistic analysis, the progression of fault into different risk levels is accurately traced which provides the advantage in early deployment of remedial actions or safety measure to confine the potential impact of fault.

6.4 Conclusion

A new approach based on SOM and probabilistic analysis has been proposed to increase the sensitivity of fault detection. This approach is verified using two experimental systems. The results from both systems have shown high sensitivity and accuracy of the proposed approach in fault detection and root cause identification. In addition, this new approach is able to determine the risk of fault as an assessment of the potential impact to the system. The main novelties of this technique lie in the fact that it is able to explore the nonlinear latent features of the high-dimensional process data samples and present these features on a 2D map in a topologically order that is easy to be interpreted. In addition, the proposed technique is also able to provide a powerful visualization of the dynamic process operation as a dynamic trajectory on a 2D map. This makes complex process system monitoring more intuitive and accessible. Moreover, in this work, the transition between normal operation and faulty operation is broken into multiple operating states to allow for a more refined classification of the risk of operation. Subsequently, this classification could lead to more robust decision making to determine the most effective safety measures. Finally, an Event tree analysis based framework is also proposed as basis for future development of the proposed technique. This framework will serve as an efficient means of assessing the effectiveness of the control actions or safety measures in minimizing the operational risk of process system.

7 Risk-based process system monitoring using Self-Organizing Map integrated with Loss Functions

Abstract

Conventional dynamic risk assessment technique does not consider the effect of nonlinear interaction among process variables for operational risk estimation. Thus, this type of technique fails to provide a realistic estimation of the operational risk of complex industrial processes. To address this issue, a multivariate risk-based process monitoring technique is proposed. This technique takes advantage of the powerful nonlinear dimensionality reduction and visualization power of the self-organizing map to identify the origin and propagation path of the fault. Through integration with the inverted normal loss function, a robust estimation of the hazard potential and operational risk of process operation can be achieved. The proposed technique is tested with two fault conditions in the benchmark Tennessee Eastman chemical process. The results have shown promising performance.

Keywords: Self-organizing map, process monitoring, loss quantification, real-time operational risk assessment

7.1 Introduction

Modern industrial processes are complex systems designed to handle a number of different tasks simultaneously. Each component of these systems has to perform a certain function while seamlessly interacting with other components to achieve the production of high-quality end products. To ensure continuously safe operation, a set of crucial process variables are monitored in real-time to determine the operating states of the system. A major challenge in process monitoring is the identification of the highly non-linear relationship between monitored variables. Once a fault condition is introduced into the system, these relationships can act as gateways for fault propagation leading to multiple upsets. These upsets, if not counteracted promptly, can result in significant process losses. To minimize these losses, it is necessary to develop a robust process monitoring technique that is capable of timely detection and accurate diagnosis of the abnormality; meanwhile, such a technique should also provide a real-time risk assessment of the fault to facilitate decision-making at an operational level.

Multivariate statistic-based techniques have received the most success in process monitoring of complex industrial processes. In these techniques, the process variables are projected into a low-dimensional feature space to form a new set of latent variables.[2-4,7,33,66] The behaviour of the latent variables can be easily analysed in low-dimensionality for abnormality detection and diagnosis. Two of the most extensively applied statistical methods are the Principal Component Analysis (PCA) and Independent Component Analysis (ICA).[7-9,67] In PCA, the projection from the process variables to the latent variables is achieved by using a set of orthogonal projection vectors. The orthogonal projection vectors represent the directions of most Gaussian variability in the monitored data. In addition, due to the orthogonal transformation, the cross-correlations (covariance) between the process variables are removed as well.[69,91] In this respect, the process variables are linearly related to a smaller number of independent Gaussian latent variables.[83] On the other hand, ICA is developed based on the assumption that the latent variables should be as non-Gaussian as possible. This requires all the high-order cross-correlation between the process variables to be removed.[16,73] The projection weight vectors for ICA are not orthogonal and represent the directions of most non-Gaussian variability in the monitored process data. From this perspective, ICA linearly relates the process variables to a group of independent non-Gaussian latent variables. For fault detection, two monitoring statistics have been developed for the latent variables of PCA and ICA, namely the Hotelling's statistic (T^2) and the Squared Prediction Error (SPE) [6]. During online monitoring, a fault condition introduces external disturbance into the process and disrupts the correlation structure between the process variables. The T^2 statistic is able to detect the breakdown of the correlation structure and SPE quantifies the magnitude of fault.[132] For fault diagnosis, a multivariate contribution chart is generated based on T^2 and SPE statistics. Process variables having high contribution to the large increase in the monitoring statistics are identified as root-cause variables. PCA and ICA represent two extreme cases of feature extraction. PCA only retains the Gaussian features while ICA merely considers the non-Gaussian behaviour of the process. In complex industrial processes, the latent process variation may comprise of both Gaussian and non-

Gaussian features. In addition, the relationship between the process variables and latent variables is extremely nonlinear due to the intricate variable interaction. These two conditions have significantly limited the capability of PCA and ICA in early fault detection and diagnosis.

To relax the limitation of PCA and ICA, self-organizing map (SOM) based fault diagnosis technique has been proposed by Yu, et al.[109]. Self-organizing map serves as a powerful non-linear feature extraction and visualization tool for process monitoring. For SOM-based process monitoring, the process data samples are captured by a layer of neurons on a two-dimensional (2D) map. As compared to PCA/ICA which has the same projection weight vector for all process data samples, each neuron of the SOM has a different set of projection weight vector. This forms the basis for non-linear projection. SOM has to be trained prior to online monitoring. The training data consists of a batch of normal process data and a batch of random faulty data.[109] In the training process, the weight vectors of the neurons are adjusted until all the process data samples are captured by a predefined number of neurons. In the meantime, the neurons on the SOM are also self-organized according to the similarity of their weight vectors. After training, the neurons corresponding to both the normal data and random faulty data form two topologically ordered clusters representing the 2D features of the normal and faulty process. During online monitoring, the online data samples are fed to the SOM per sample interval. A 2D dynamic trajectory connecting the neurons being hit by the online data samples is used to visualize the process operation. When the process is operating normally, the trajectory stays in the normal cluster. In the case of a fault, the process data samples diverge from normal pattern and get captured by neurons farther away from the normal cluster. Consequently, the trajectory deviates away from the normal cluster and a fault condition is detected if the dynamic trajectory enters the faulty cluster. For fault diagnosis, a multivariate contribution plot is generated based on the contribution of each process variable to the deviation of trajectory.

Upon detection of fault and identification of the high contributing variables, the process losses incurred by the abnormality of the high contributing process variables are to be quantified. Several researchers have proposed different types of loss functions to quantify process losses. Zadakbar, et al.[133] and Hashemi, et al.[134] provided a comprehensive review of these loss function and discussed in detail their application in dynamic loss modelling. Hashemi, et al.[135] proposed the use of an inverted beta loss function to model the loss concerning a temperature surge in a continuous stirred reactor tank. In general, these loss modelling methods associate the deviation of the process variables from target value with the process losses. However, the research work of Hashemi, et al.[134] and Hashemi, et al.[135] focused on modelling the process losses considering the deviation of a single process variable; the interaction between process variables that could lead to multiple upsets and significant increase in overall risk is not taken into account. In this work, the loss function model is integrated with SOM to estimate the process operational risk in a multivariate context. The inverted normal loss function (INLF) proposed by Spiring[136] is adopted since the shape parameter of the INLF can be easily adjusted to allow the maximum loss to be reached within a certain limit of

process deviation. This flexibility is particularly beneficial when being used to quantify loss for process with several operational constraints.

The loss functions provide an estimate of the hazard potential of each high contribution process variable based on their deviations. The operational risk of the process is computed as the product of the hazard potential and the likelihood of fault occurrence.[124,134,137] In the present study, a histogram approximation of the probability distributions of the normal process data and random fault data is constructed on the 2D SOM map based on the number of data samples each neuron receives during the training process. A mixing parameter for each of the probability distribution is also estimated to obtain a normalized joined probability distribution on the SOM map. During online monitoring, once a neuron captures an online data sample, the likelihood of fault occurrence is updated using Bayes' theorem. Meanwhile, the dynamic contribution of each process variable to the deviation of the dynamic trajectory is also computed in real-time. For identification of high contributing process variables, upper and lower control limits are determined for the dynamic contribution of each process variable. When a fault is detected, each process variables with high dynamic contribution (breaching control limits) is assigned a loss function to model univariate loss. Subsequently, the process loss is determined as the maximum loss among the univariate losses. Finally, the operational risk of the process is calculated as the product of the maximum loss and the instantly updated likelihood of fault occurrence.

The remainder of this article is organized as follows: A brief review of the basic principles of SOM is provided in the Background section. The derivation of the risk-based process monitoring technique is illustrated in detail in the Methodology section. In the case study section, the effectiveness of the proposed technique is verified using two pre-programmed fault conditions in the benchmark Tennessee Eastman chemical process. The major conclusions and the contributions of this work are summarized in the Conclusion section.

7.2 Background

Self-organizing map was first proposed by Kohonen[138] as a type of unsupervised vector quantization technique. Self-organizing map is able to discover the nonlinear latent features from high dimensional data. These low-dimensional features are presented in the form of a layer of topologically ordered neurons on a 2D map. A simple example of the 2D SOM is shown in Figure 7-1.

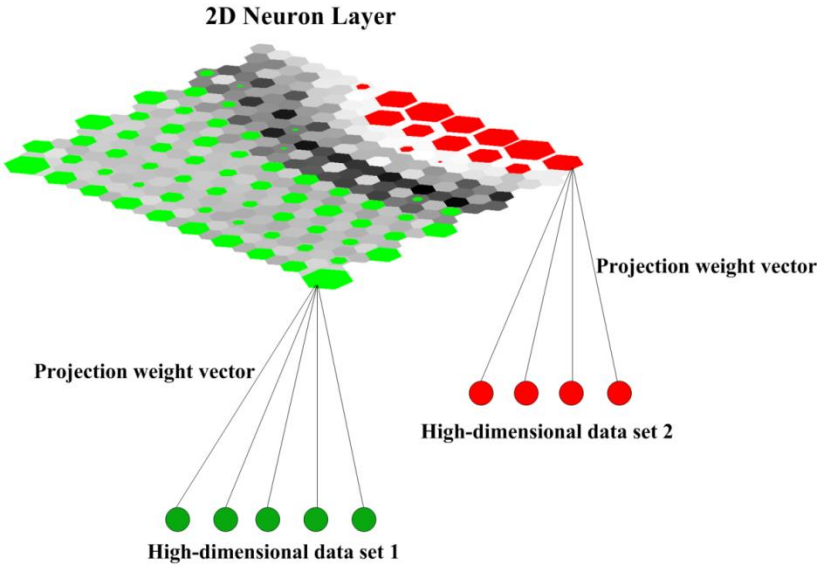

Figure 7-1: SOM feature extraction.

The ability of the SOM to discover the latent feature is measured by a quantity known as the quantization error. Consider a small data batch $\mathbf{X} \in \Re^{N\times d}$ captured by a neuron with a weight vector $\mathbf{w}_j \in \Re^{1\times d}$. The quantization error of this particular neuron is defined as.

$$e_j = \sum_{t=1}^{N} (\mathbf{x}^t - \mathbf{w}_j)^{\mathrm{T}} (\mathbf{x}^t - \mathbf{w}_j) \tag{7.1}$$

where $\mathbf{x}^t \in \mathbf{X}$ is the t^{th} data sample of the data set. The overall quantization error of the SOM is the summation of the quantization errors of all neurons.

$$E = \sum_{j=1}^{M} e_j \tag{7.2}$$

where M is the total number of neurons on the map. This error is also used as an indicator to show the training progress of the SOM. A single iteration of training consists of three steps: competition, cooperation and adaptation. In the competition step, the weight vectors of the neurons are first linearly initialized along the eigenvectors of the training data set and the number of neurons is set to $M = 5 \times N^{0.54321\ 139}$. The weight vectors are then compared with a single training data sample. The neuron that has the lowest quantization error is declared as the wining neuron or the Best Matching Unit (BMU).

$$\mathbf{w}_{BMU} = \arg\min_{\mathbf{w}_j} \left\| \mathbf{w}_j - \mathbf{x}^t \right\|, \forall j \in \{1:M\}, \forall t \in \{1:N\} \tag{7.3}$$

Subsequently, in the cooperation step, the direct neighbourhood neurons of the BMU are identified. Finally, the weight vectors of the BMU and its neighbours are selectively tuned to minimize the quantization error. The tuning function is expressed as.

$$\mathbf{w}_j(t+1) = \mathbf{w}_j(t) + \alpha(t) d_j(t) \left[\mathbf{x}^t - \mathbf{w}_j(t) \right] \tag{7.4}$$

where $\alpha(t)$ is the tuning rate and $d(t)$ is the exponential neighbourhood function. $\alpha(t)$ decreases exponentially over iteration resulting in a more refined tuning towards the end of training process.

$$\alpha(t) = \alpha_0 \exp\left(\frac{-t}{\lambda}\right) \tag{7.5}$$

where α_0 is the initial learning rate and λ is the time constant which is determined as.

$$\lambda = \frac{N}{\sigma_0} \tag{7.6}$$

where σ_0 is the radius of the map. It is computed as the Euclidean distance between the coordinates of the outmost neuron and the centre neuron.

$$\sigma_0 = \left\| \mathbf{c}_{outmost} - \mathbf{c}_{centre} \right\| \tag{7.7}$$

It is noted that on 2D map, the coordinates of each neuron is expressed as $\mathbf{c}_j = [c_j^1 \; c_j^2]$. $\mathbf{c}_{outmost}$ denotes the coordinate of the outmost neuron while $\mathbf{c}_{centre}$ represents the coordinate of the central neuron. On the other hand, $d(t)$ is maximized at the BMU and decays exponentially with the distance from the BMU.

$$d(t) = \exp\left(\frac{\left\| \mathbf{c}_j - \mathbf{c}_{BMU} \right\|^2}{2\sigma(t)^2} \right) \tag{7.8}$$

$$\sigma(t) = \sigma_0 \exp\left(\frac{-t}{\lambda}\right) \tag{7.9}$$

Where c_{BMU} is the coordinate of the best matching unit and $\sigma(t)$ is the radius of the neighbourhood. This means that the neurons that are farther away from the BMU are updated at a much lower rate. In addition, the neurons that are outside the radius of neighbourhood are skipped completely. As a result, the weight vectors of the BMU and its neighbours gradually become more similar to the input data samples. Conversely, the similarity between the weight vectors of the neurons farther away and the input data sample decreases over time. This type of differential tuning leads to topological ordering of the neurons. The training process stops until the overall quantization error falls below a certain threshold or the maximum number of training iterations is reached. After training, the topologically ordered neurons form a 2D pattern which corresponds to the low-dimensional latent features of the training data samples, such as the example shown in Figure 7-1. In this work, the SOM toolbox developed by Vesanto, et al.[139] is utilized for SOM training and feature extraction.

In addition to feature extraction, each neuron captures a number of data samples from the training data set. In fact, the weight vector of the neuron represents the expected value of the data samples captured. This mechanism provides powerful density estimation for large and high-dimensional data. A histogram can be generated at each neuron depending upon how many data samples it captures. The neurons that capture a large amount of data samples have high histogram and represent a region of high data concentration. This region also contains the most significant variation of the high-dimensional data. By normalizing the heights of the histogram for all the neurons, it is able to obtain an approximation of the probability density function of the data. For complex process monitoring, a major advantage of such density estimation method is that there is no assumption on distribution types (Gaussian or non-Gaussian). The probability density function is completely self-organized by the neurons. Therefore, it represents more realistic distribution of the process data.

7.3 Methodology

7.3.1 Estimation of Failure Probability

The SOM is first trained with 1000 samples of normal process data and 1000 samples of random faulty data. The random fault data prevents the neurons representing the normal process data from occupying the entire 2D map, so that a clear boundary is formed around the 2D normal process feature. The random fault data is sampled from a Laplace distribution with zero mean and a scale factor of 100. An example of the trained SOM is demonstrated in Figure 7-2(a), where the green pattern represents the normal process feature and the red cluster characterizes the feature of the random faulty data. Each neuron is represented as a hexagon and the size of the hexagon is proportional to the number of data samples captured. For process monitoring, the online process data is fed to the trained SOM per sample interval. The online data sample is then compared with the weight vectors of the neurons. The neuron having the smallest quantization error captures the data sample and is marked in blue. By connecting the neurons that capture each online data sample, a dynamic trajectory representing the dynamic behaviour of the process is shown in Figure 7-2(b). When the process is operating normally, normal process data samples are generated and captured by neurons in the green cluster. A fault condition introduces abnormal variation into the system leading to generation of data samples with very different features. These data samples are then captured by the neurons farther away from the green cluster and force the dynamic trajectory to diverge from normal. The benefit of this process monitoring technique is that it provides a direct visualization of the process operation.

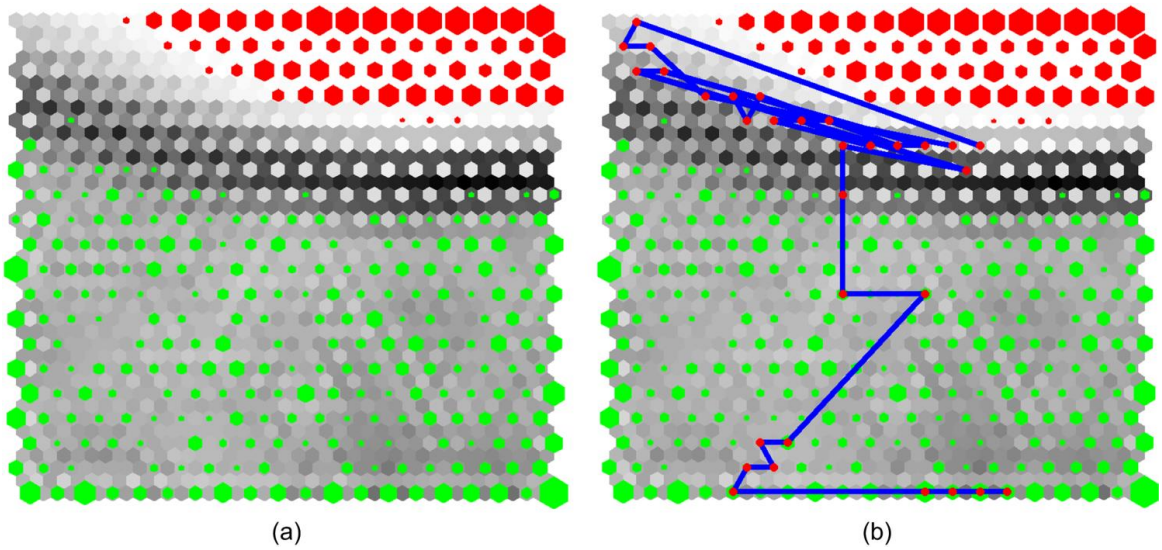

Figure 7-2: (a) SOM trained with normal process data and random fault data; (b) SOM based process monitoring.

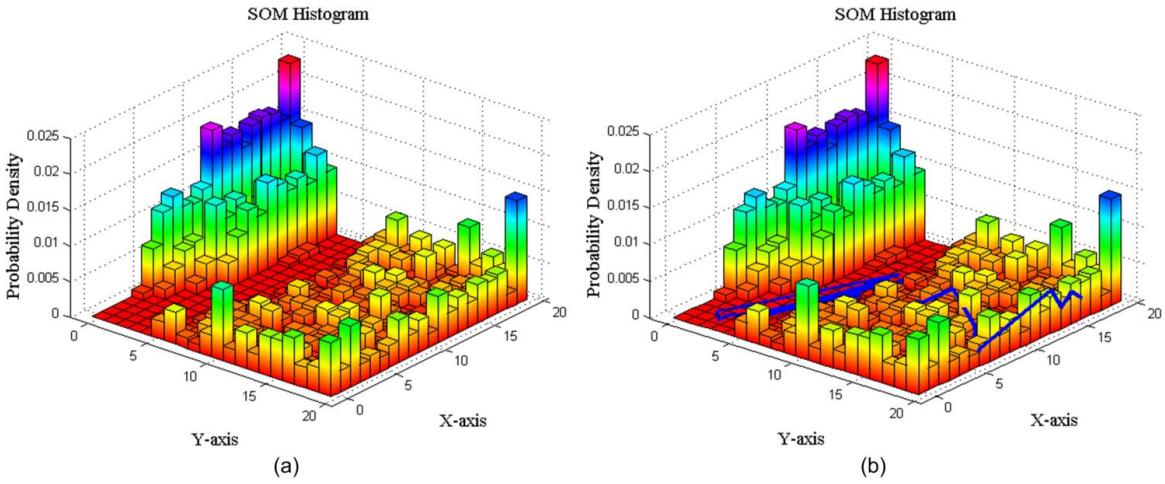

Figure 7-3: Histogram approximation of the probability density function.

To approximate the probability density function of the process data distribution, a histogram is constructed for each neuron based on the number of data samples they capture during the training process. Subsequently, the heights of the histograms are normalized such that they sum up to 1. The normalized histogram represents the probability of a neuron will capture online data samples during process operation. Those neurons that capture a large amount of training data retain the most significant variation of the process and have high probability of capturing online data samples. The histogram representation of Figure 7-2(a) is demonstrated in Figure 7-3(a). In total, there are 400 neurons arranged on the 20 by 20 SOM map. Each neuron is also assigned a coordinate on the map. For example, the first neuron in the bottom left corner of Figure 7-2(a) is given a coordinate (0, 20) as shown in Figure 7-3(a). Likewise, the bottom right corner neuron is given a coordinate (20, 0).

It is observed that the normal neurons and faulty neurons form two regions of high probability density. These two regions can be considered as two probability distributions.

The joined probability distribution on the 2D map is obtained by mixing these distributions. The mixing parameters for these two probability distributions are estimated as the ratio between the total number of training data samples captured in the red region or green region and the total number of data samples.

$$P(k_1) = \frac{N_{k1}}{N} \tag{7.10}$$

$$P(k_2) = \frac{N_{k2}}{N} \tag{7.11}$$

Where $N_{k1}$and N_{k2} are the total number of captured data samples in the normal (green) region and fault (red) region, respectively. N is the total number of data samples. Additionally, the probability of each neuron capturing data samples is calculated as.

$$P(neu_j) = \frac{n_j}{N} \tag{7.12}$$

Where n_j is the number of data samples neuron j, neu_j, receives during the training process. If neuron j is in either the green or red region, the conditional probability of neuron j capturing data samples when the system is operating normally or abnormally is given as.

$$P(neu_j \mid k_1) = \frac{n_j}{N_{k1}} \tag{7.13}$$

$$P(neu_j \mid k_2) = \frac{n_j}{N_{k2}} \tag{7.14}$$

During online monitoring, suppose that the online data sample $\mathbf{x}^t$ is captured by neuron j, the posterior probability of the data sample belonging to the normal (green) region is updated using Bayes' theorem.

$$P(k_1 \mid \mathbf{x}^t) = \frac{P(k_1)P(neu_j \mid k_1)}{P(neu_j)} \tag{7.15}$$

This is the probability of the process operating normally at sample interval t. The posterior probability of fault is therefore the complementary of Equation (7.15).

$$P_t(\text{fault}) = 1 - P(k_1 \mid \mathbf{x}^t) \tag{7.16}$$

As shown in Figure 7-3(b), the fault condition forces the dynamic trajectory to diverge from the normal region. In the interim, it is also noticed that the height of the histograms along the dynamic trajectory decreases gradually indicating diminishing likelihood of

normal operation. Eventually, the probability of normal operation becomes almost zero when the trajectory moves into the red region.

7.3.2 Identification of High Contributing Process Variables

The powerful nonlinear dimensionality reduction ability of SOM allows the process variation to be presented as a 2D dynamic trajectory on the map. As the fault condition propagates, the process generates data with very dissimilar features compared to that of the normal data. This leads to further deviation of the dynamic trajectory. In this regard, the deviation of the trajectory reflects the magnitude of the fault. Yu, et al.[109] proposed a dissimilarity index to quantify the magnitude of the deviation. The dissimilarity index calculates the Euclidean distance between the coordinates of the neuron, neu_j and the reference neuron receiving most data samples in the normal region.

$$Disim_j = \left\| \mathbf{c}_j - \mathbf{c}_{ref} \right\| \tag{7.17}$$

where $\mathbf{c}_j$ is the coordinate of the neuron, neu_j. $\mathbf{c}_{ref}$ is the coordinate of the neurons, neu_{ref}, in the normal region that receives most data samples during training process. This particular neuron is used as a reference point of normal operation since it captures the most significant variation of the process operation. This index makes it possible to further reduce the dimensionality of the process monitoring to one; the process variation is expressed as the variation in the Euclidean distance between neu_j and neu_{ref}. From this point of view, the online projection from the process data to the dissimilarity index is expressed as.

$$Disim_j = \mathbf{x}_j^t \mathbf{v}_{neuj}(t) \tag{7.18}$$

where $\mathbf{x}_j^t$ is the data sample captured by neu_j at sample interval *t* during online monitoring. $\mathbf{v}_{neuj}(t)$ is known as the dynamic loading vector for neu_j and each entry of this vector indicates the dynamic contribution of each process variable in $\mathbf{x}_j^t$ to the instant variation of the process [109]. The dynamic loading vector is obtained by solving the following optimization problem.

$$\hat{\mathbf{v}}_{neuj}(t) = \underset{\mathbf{v}_{neuj}(t)}{\arg\min} \sum_{t=1}^{n_j(t)} \left[Disim_j - \mathbf{x}_j^t \mathbf{v}_{neuj}(t) \right] + \delta^2 \left[\mathbf{v}_{neuj}(t) \right]^{\mathbf{T}} \mathbf{v}_{neuj}(t) \tag{7.19}$$

where δ is the L2 regularize coefficient, $n_j(t)$ is the total number of samples neuron *j* captures at sample interval *t* during online monitoring. The regularizer adds a small value in the diagonal of the pseudo inverse $\left[\left(\mathbf{x}_j^{1:n_j(t)} \right)^{\mathbf{T}} \mathbf{x}_j^{1:n_j(t)} \right]^{-1}$ to make it well-conditioned. The regularizer coefficient is determined using the generalized cross-validation approach (GCV).[140] The GCV function is expressed as.

$$G(\eta) = \frac{\frac{1}{n_j(t)} \left\| \mathbf{I} - \mathbf{A}(\eta)\mathbf{Disim} \right\|^2}{\left\{ \frac{1}{n_j(t)} \mathrm{Trace}\left[\mathbf{I} - \mathbf{A}(\eta) \right] \right\}^2} \tag{7.20}$$

where $n_j(t) = \delta^2$, **Disim** is a vector that consists of $n_j(t)$ number of $Disim_j$ and,

$$A(\eta) = \mathbf{x}_j^{1:n_j(t)} \left[\left(\mathbf{x}_j^{1:n_j(t)} \right)^{\mathbf{T}} \mathbf{x}_j^{1:n_j(t)} + n_j(t)\eta \mathbf{I} \right]^{-1} \left(\mathbf{x}_j^{1:n_j(t)} \right)^{\mathbf{T}} \tag{7.21}$$

The optimal δ is determined using the grid search method.

$$\begin{aligned} \hat{\eta} &= \arg\min_{\eta} G(\eta) \\ \delta^2 &= n_j(t)\hat{\eta} \end{aligned} \tag{7.22}$$

Expression (7.19) has a closed-form solution which is obtained by setting its first derivative to zero.

$$\hat{\mathbf{v}}_{neuj}(t) = \left[\left(\mathbf{x}_j^{1:n_j(t)} \right)^{\mathbf{T}} \mathbf{x}_j^{1:n_j(t)} + \delta^2 \mathbf{I} \right]^{-1} \left(\mathbf{x}_j^{1:n_j(t)} \right)^{\mathbf{T}} \mathbf{Disim} \tag{7.23}$$

When the process system is operating normally, each process variable has a steady contribution to the operation states. A fault condition disrupts the correlation between process variables and causes abnormal variation in dynamic contribution of a certain group of variables. These variables are closely related to the fault condition and are also responsible for propagating the fault. To identify these process variables, upper and lower control limits are assigned to the dynamic contribution of each process variable. These limits are set as three standard deviations above and below the mean value of the first 1000 samples of normal dynamic contribution. The identified high contributing process variables are then integrated with loss functions to determine the maximum possible loss of the process in real-time. The major advantage of quantifying process losses based on the deviations of the high contributing process variables is that proper safety measures or remedial actions can be directly targeted at the origin of fault; the process can be brought back to safe state with minimum delay and losses by eliminating the origin and severing the propagation path of the fault.

7.3.3 Estimation of the Operational Risk

The operation of complex industrial processes is often subjected to multiple constraints to prevent catastrophic failure. These constraints are set as high and low shutdown limits for critical process variables. In this case, the process loss starts to increases if the high contributing process variables exceed their high or low normal operation limits. The maximum process loss is reached if any of the identified variables breaches the shutdown

limit. The loss function used for this study to quantify the process loss is the inverted normal loss function which takes the following form [136].

$$L(x_d^t) = EML\left[1 - e^{-\frac{(x_d^t - T_d)}{2\gamma^2}}\right] \tag{7.24}$$

where $x_d^t \in \mathbf{x}^t$ is the d^{th} identified high contributing process variable in online data sample $\mathbf{x}^t$. T_d is the high or low operational limit for x_d. $\gamma = \Delta/4$ is the shape parameter defining at which value of the process variable the maximum loss is reached. Δ is the difference between the shutdown limit and the high or low normal operation limit. EML is the expected maximum process loss. Expression (7.24) calculates the univariate loss of the identified process variable at each sample interval. The maximum possible loss of the process in real-time is determined by taking the maximum real-time univariate loss.

$$L_t(\text{Process}) = \max_d L(x_d^t) \tag{7.25}$$

Subsequently, the real-time operational risk of the process is calculated as the product of the likelihood of fault and the maximum possible loss.

$$Risk_t = L_t(\text{Process}) \times P_t(\text{fault}) \tag{7.26}$$

As compared to the univariate risk estimation method developed by Zadakbar, et al.[133], Hashemi, et al.[134] and Hashemi, et al.[135], the proposed methodology has the following strengths:

(1) The 2D dynamic trajectory of SOM provides a direct visualization of the states of process operation;
(2) The probability of fault is calculated by considering the nonlinear relationship between the process variables;
(3) In addition, the probability of fault is also determined without assuming any distribution type of the data; thus it better reflects the realistic situation.
(4) The losses associated with all the high contributing process variables are considered for maximum possible loss estimation;
(5) The process operational risk is estimated in a multivariate context;
(6) The identification of high contribution process variables also reveals the fault propagation path and allows the safety measures to be directly targeted at the origin of fault, so that the process can be brought back to safe region as soon as possible;

The logical flow diagram of the proposed risk-based process monitoring technique is shown in Figure 7-4.

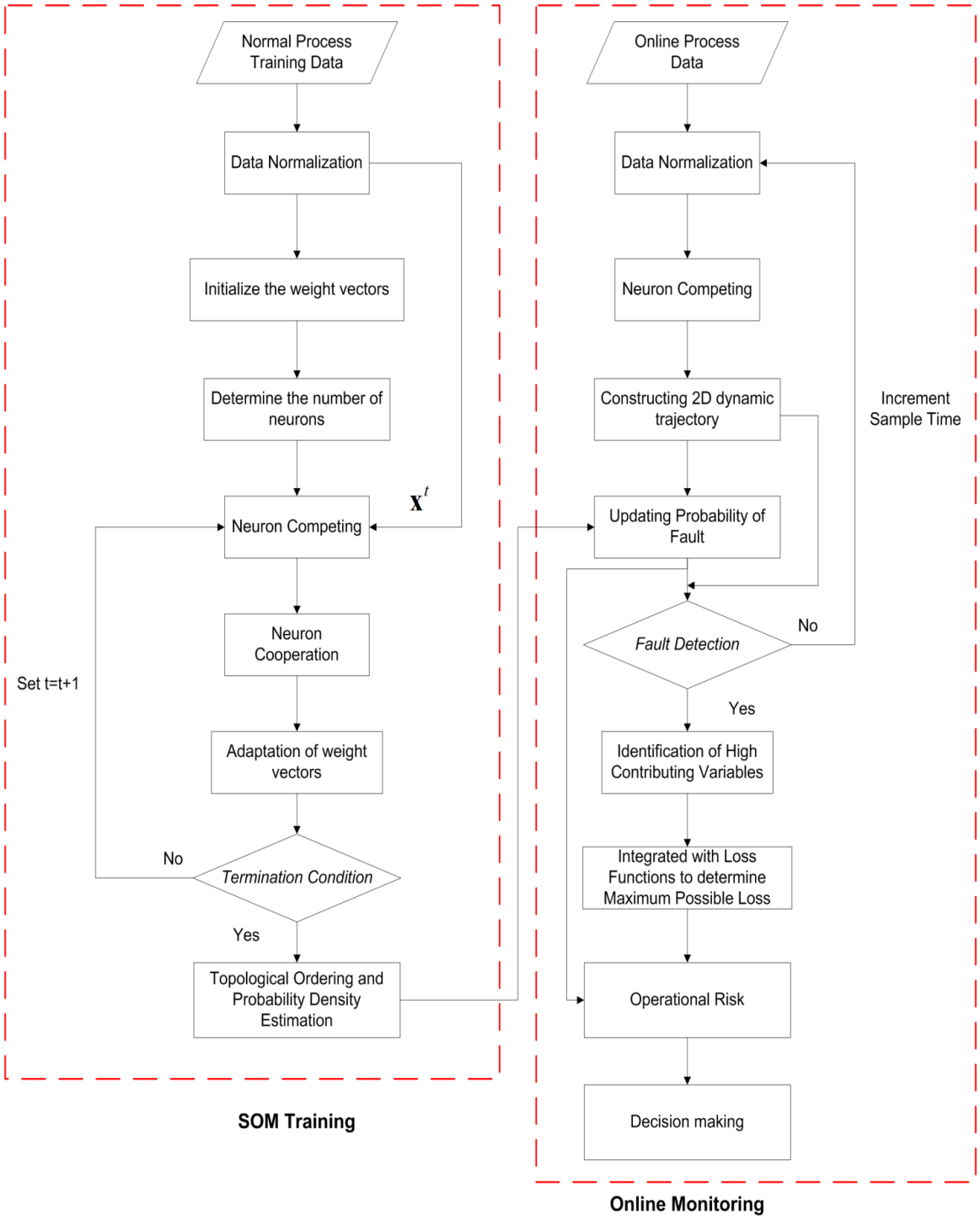

Figure 7-4: Logic flow chart of the proposed risk-based process monitoring technique.

7.4 Case Study

In this section, the effectiveness of the proposed risk-based process monitoring technique is tested with a Simulink simulation of the benchmark Tennessee Eastman

chemical process. The simulation adopts the decentralized control strategy to achieve a closed-loop stable simulation of the process.[86] The process flow diagram of the chemical plant is shown in Figure 9-1. The detail description of the process can be found in the work of Downs, Vogel[32].

In total, there are 41 measured process variables in the process. Twenty-two of these variables are monitored to determine the operating condition of the process system. These monitored variables are listed in Table 9-1. In addition, 15 fault conditions have been included in the simulation package. These fault conditions have been widely used by the statistical process monitoring community to verify and compare various techniques [141]. In this study, 2 of these fault conditions are used to verify the proposed technique. The process is monitored for 7200 sample intervals and the fault conditions are introduced at sample interval 3000.

Table 7-1. Tested Fault conditions

Fault ID	Fault description	Signal Type
IDV6	Feed loss in Feed A (stream 1)	Step
IDV13	reaction kinetics	Slow drift

One-thousand normal process data samples and the same number of random faulty samples[***] are generated to train the SOM for each of the fault conditions. The standard data normalization procedure is then applied to the training data samples. The processed training data has zero-mean and unit variance. The process losses are divided into two major categories: material loss and shutdown loss. Material loss is associated with process variables whose abnormal variation will cause degradation of productivity. The shutdown loss is the maximum loss the process suffers if any of the critical process variables breaches its shutdown limit. The maximum value for process losses can be obtained from historical data. In this case, for simplicity of demonstration, the maximum material loss is set to $300 dollars and the maximum shutdown loss is set to $700 dollars.

7.4.1 IDV6: Feed loss in Feed A

In this fault condition, a step loss is introduced to feed A of process. The dynamic trajectory of and the dynamic probability of fault of this fault condition is shown in Figure 7-5.

[***] The faulty data samples are generated according to method outline in the first paragraph of section 7.3.1. The normal data samples are collected directly from the TEP simulation under normal operating condition.

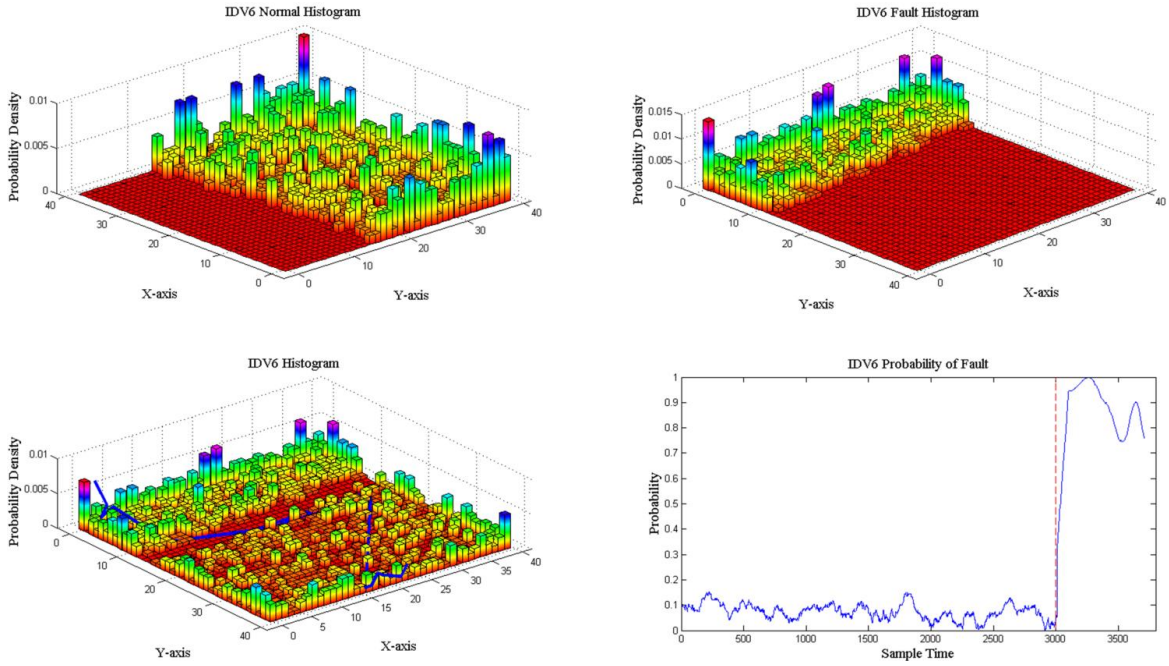

Figure 7-5: Fault detection results of IDV6.

It is readily observed that the probability of fault increases drastically right after the fault is introduced at sample interval 3000; this demonstrates high sensitivity of the proposed technique in capturing abnormality in complex process system due to its ability to discover the non-linear features of the process. The simulation was terminated at sample interval 3709 when the stripper level drops below the shutdown limit. Next, the dynamic loading of each process variable is calculated using the method outlined in section 7.3.2 and is presented in Figure 7-6. From Figure 7-6, the high contributing process variables are identified as X1, X7, X15, X16 and X18 as they keep breaching their control limits after the fault is introduced.

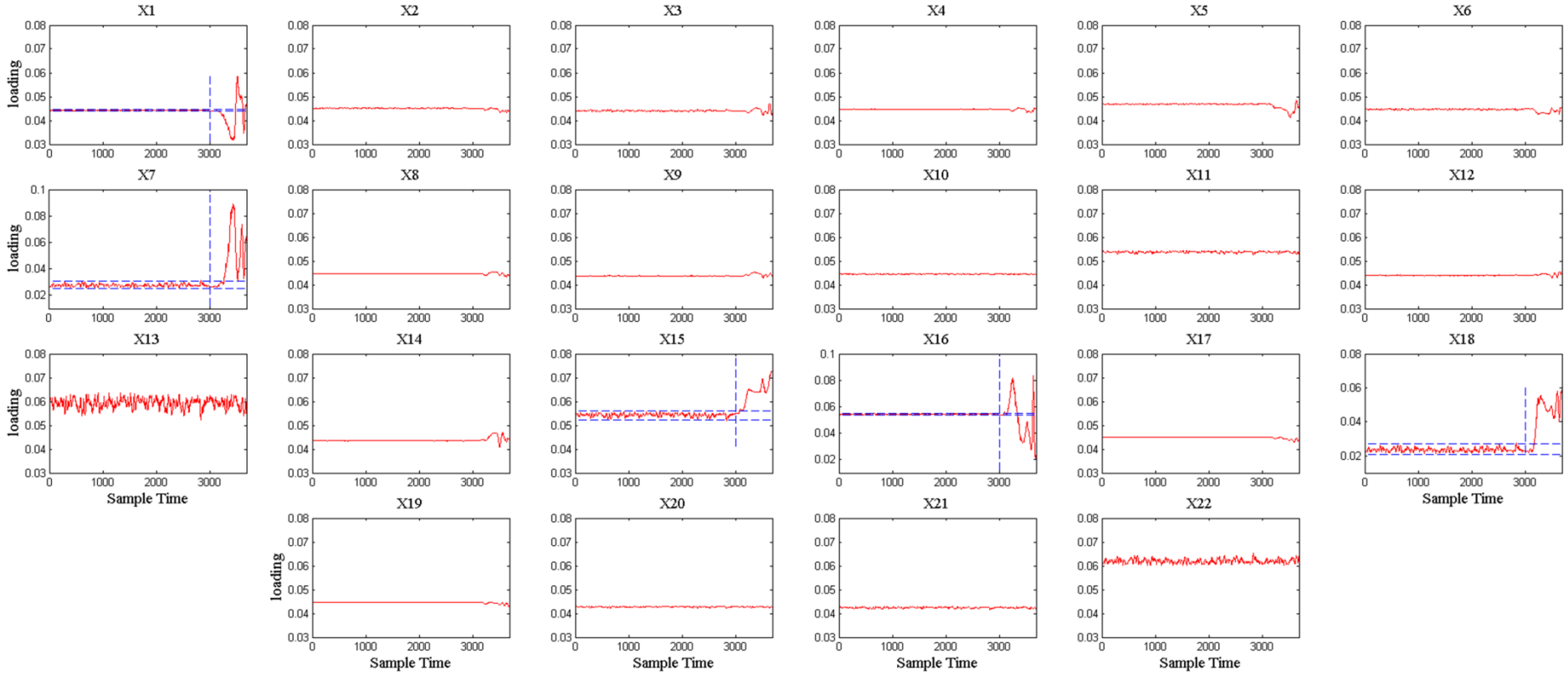

Figure 7-6: Dynamic loadings of monitored process variables for IDV6.

The root-cause of the fault is a step loss in feed A. It is correctly identified from the abnormal variation in dynamic loading of process variable X1. This fault condition is then propagated downstream to disrupt the material balance of the reaction taking place in the reactor. As a result, the reactor pressure (X7) shows abnormal behaviour as well. This fault condition continues to move downstream and eventually upsets the operation of the stripper. In particular, the stripper level decreases below the lower shutdown limit leading to shutdown of the process and termination of the simulation. Meanwhile, the steam flow (X19) and feed C (X4) of the stripper are still supplied at normal rate which causes the abnormal behaviour of the stripper pressure (X16) and temperature (X18). The operational profiles and the constraints of these identified process variables are also shown in Figure 7-7 and Table 7-2, respectively.

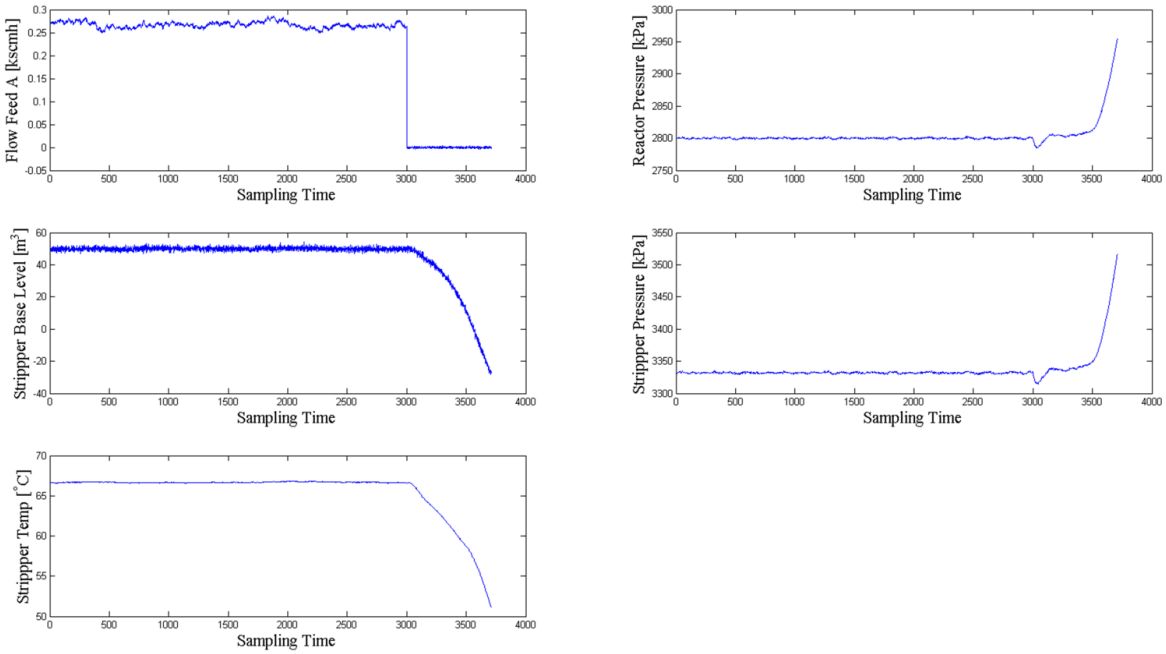

Figure 7-7: Operational profile of the high contributing process variables for IDV6.

Table 7-2: Operational constraints for identified process variables in IDV6.

Process Variable	Low operation limit	High operational limit	Low shutdown limit	High shutdown limit	Ref.	Loss Type
X1	0.1	-	-	-	Assumed	Material
X7	-	2895	-	3000	[32]	Shutdown
X15	11.8	21.3	2	24	[32]	Shutdown
X16	3.5	6.6	1	8	[32]	Shutdown
X18	-	68	-	120	Assumed	Shutdown

The univariate losses of the identified process variables with the exception of X1 are calculated using the Inverted Normal Loss Functions under the constraints listed in Table 7-2. For X1, a step loss function is used in reflecting the step fault. The losses corresponding to the deviation of the identified process variables are shown in Figure 7-8.

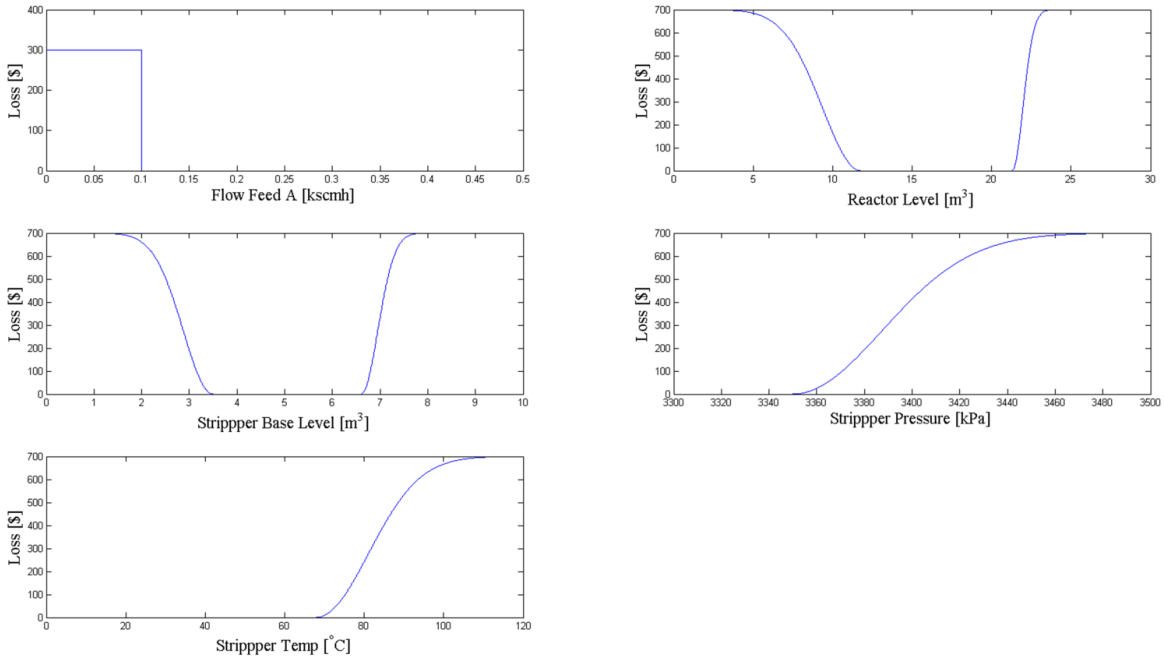

Figure 7-8: Loss functions for IDV6.

The real-time maximum possible loss of the process is determined by taking the maximum univariate loss at each sample interval. Finally, the operational risk of the process is computed using expression(7.26). The results of these calculations are shown in Figure 7-9. The proposed multivariate loss calculation provides a more robust estimation of hazard potential of process operation. For example, at sampling time 3600, if only the loss of the root-cause variable X1 is considered, the system loss is only at 300 dollars which is much lower as compared to the real loss caused by rapidly diminishing stripper level. Consequently, the operational risk is also small which will adversely impact the decision making to determine the proper safety measures. In contrast, the proposed technique is not only able to detect the fault promptly but also it is capable of identifying the fault propagation. The process operation can be brought back to safe region with minimum delay and cost.

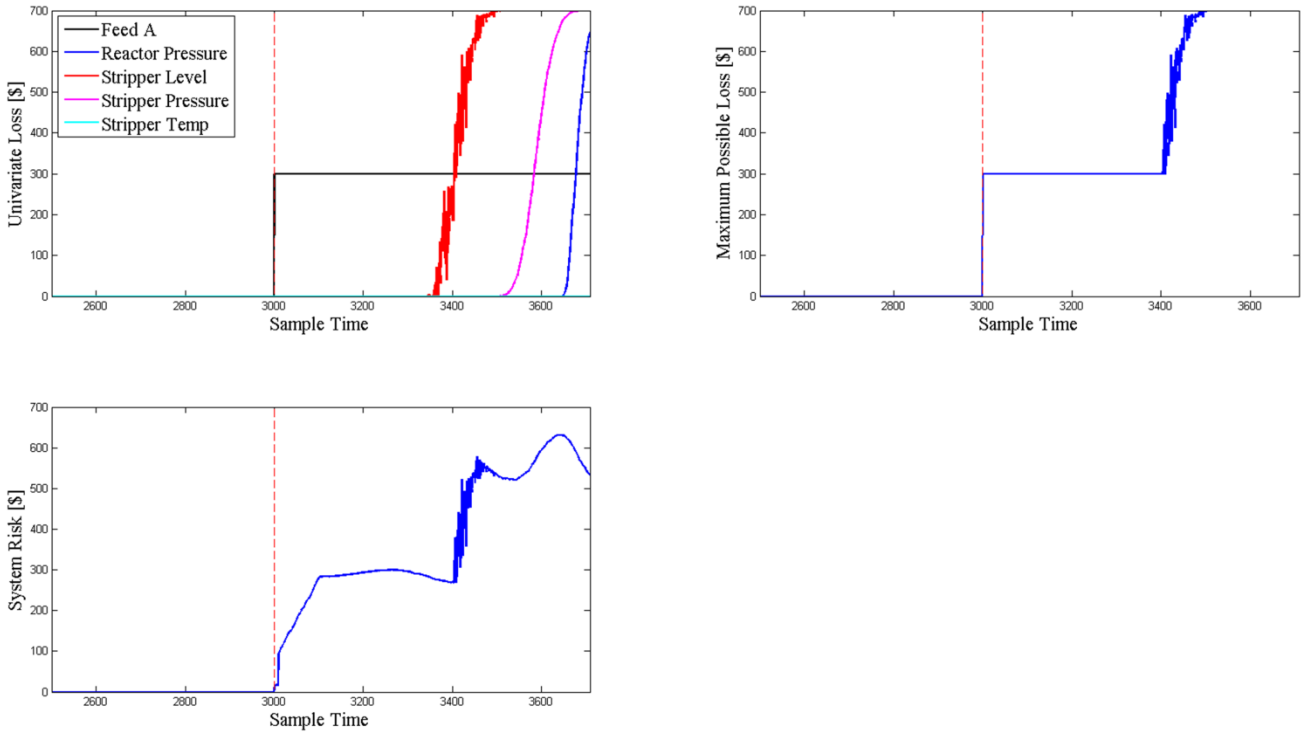

Figure 7-9: Operational risk of the TEP under fault condition IDV6.

7.4.2 IDV13: slow drift in reactor kinetics

In this case study, the fault condition is introduced as a slow drift in the reaction kinetics. The first operating unit that will be affected by this fault condition is the reactor. The dynamic trajectory and probability of fault of IDV13 are shown in Figure 7-10. As the fault condition is quite subtle at the beginning, there is a small delay in fault detection—the probability of fault starts to increase significantly only after sampling time 3290. Despite this fact, most of the abnormal behaviour from sampling time 3000 to 7200 is correctly identified.

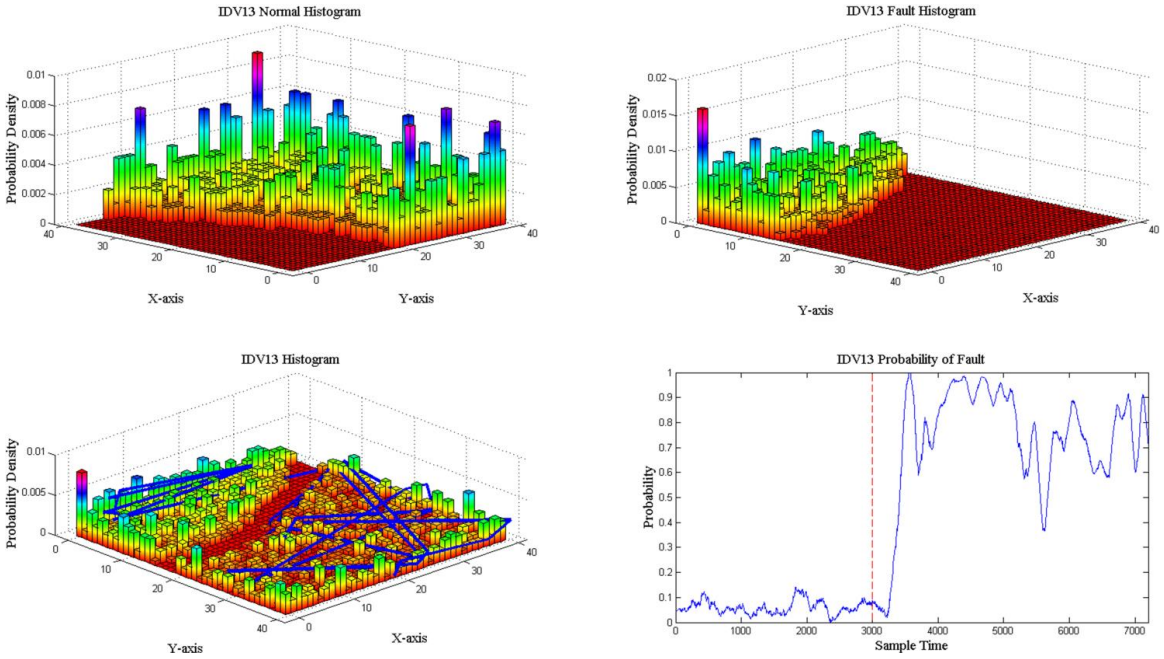

Figure 7-10: Fault detection results of IDV13.

The dynamic loading for each of the monitored variable is shown in Figure 7-11. The high contributing process variables are identified as X7, X10, X11, X16, X18 and X20. It is noted that X22 is not included into the set due to the reason that it does not link to any downstream or recycle variable of the process; the abnormal behaviour of X22 will not help propagate the fault condition. The identified process variables also correctly represent the fault propagation path. The slow drift in reactor kinetics results in change of reactor pressure (X7). X7 is correctly identified as the origin of fault. In turn, this fault condition propagates to downstream operating units. Due to the change in the reaction condition, the rate of purging (X10) the inert product is also adversely affected leading to material loss. Due to the same reason, the compressor responsible for recycling the excess reactants also shows abnormality. Moving further downstream, the separator temperature is the first monitored process variable after the purge rate that links the fault propagation path to the last operating unit—the product stripper. Similar to IDV6, the stripper pressure (X16) and stripper temperature (X18) are affected as well. The operational profiles and the constraints of these identified process variables are also shown in Figure 7-12.

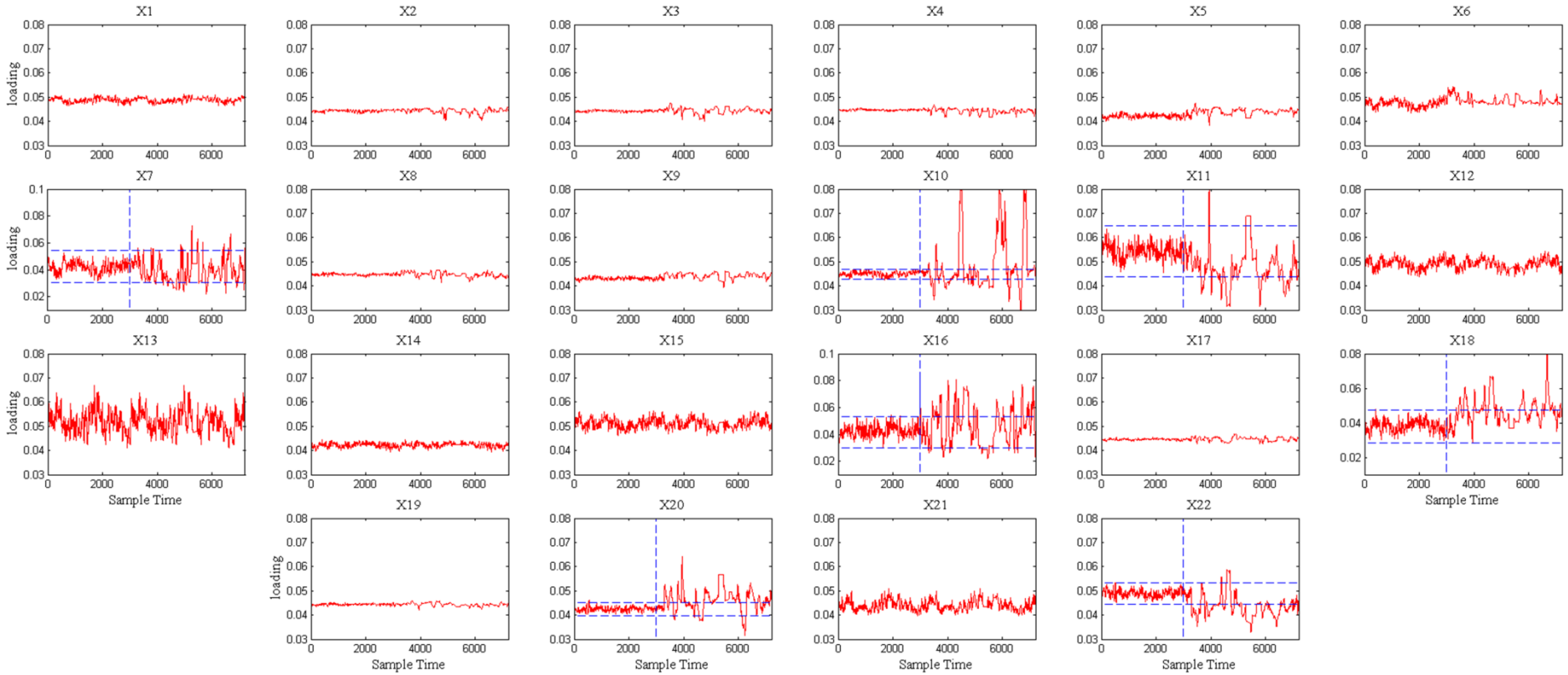

Figure 7-11: Dynamic loadings of monitored process variables for IDV13.

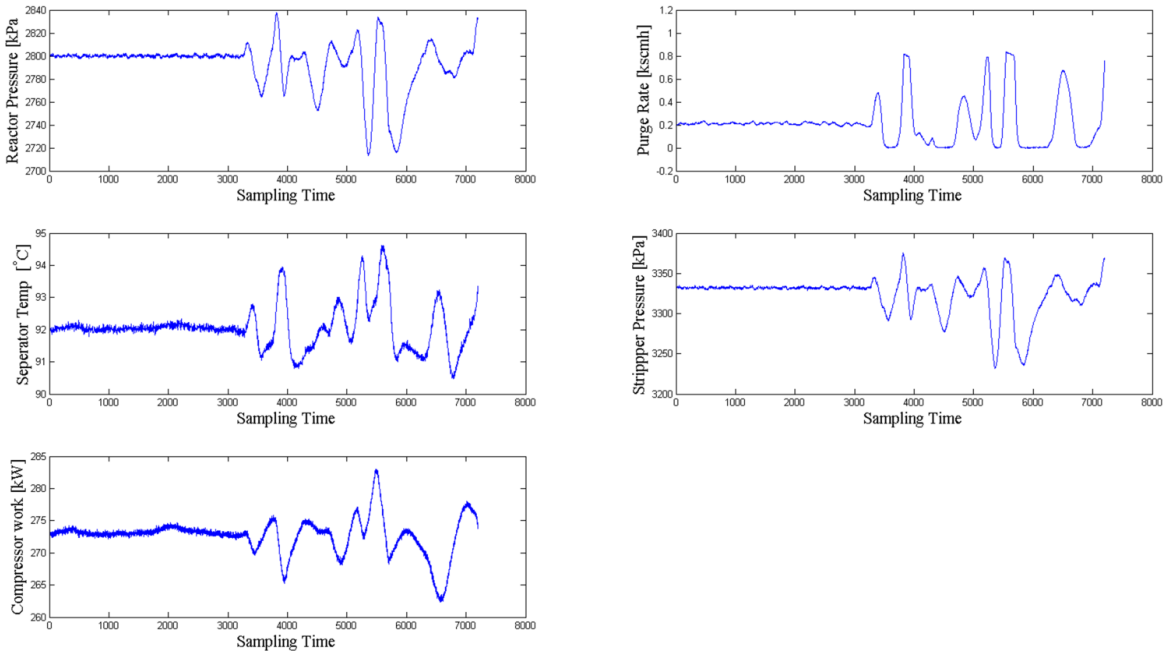

Figure 7-12: Operational profile of the high contributing process variables for IDV13.

In addition to Table 7-2, Table 7-3 summarizes the operational constraints of the X10, X11 and X20.

Table 7-3: Operational constraints for identified process variables in IDV13.

Process Variable	Low operation limit	High operational limit	Low shutdown limit	High shutdown limit	Ref.	Loss Type
X10	-	0.24	-	0.48	Assumed	Material
X11	-	93	-	100	Assumed	Shutdown
X20	260	280	220	320	Assumed	Material

Under these operating constraints, the inverted normal loss functions of the identified high contributing process variables are shown in Figure 7-13. Finally, the process loss and operational risk under IDV13 is shown in Figure 7-12. The risk is much lower than IDV6 as the process operation does not violate any shutdown limits and the decentralized closed loop control algorithm is trying to bring the system back to the set-point. The control actions are not stable due to the constant injection of the fault. This leads to the fluctuation in the process operation. This fluctuation is well reflected in the real-time risk profile. Likewise, this risk estimation is much more robust as compared to the univariate case. For instance, if only the root-cause process variable reactor pressure (X7) is considered, the operational risk of the process is zero because the fluctuation is well within the operational range. Evidently, this will seriously underestimate the risk of operation.

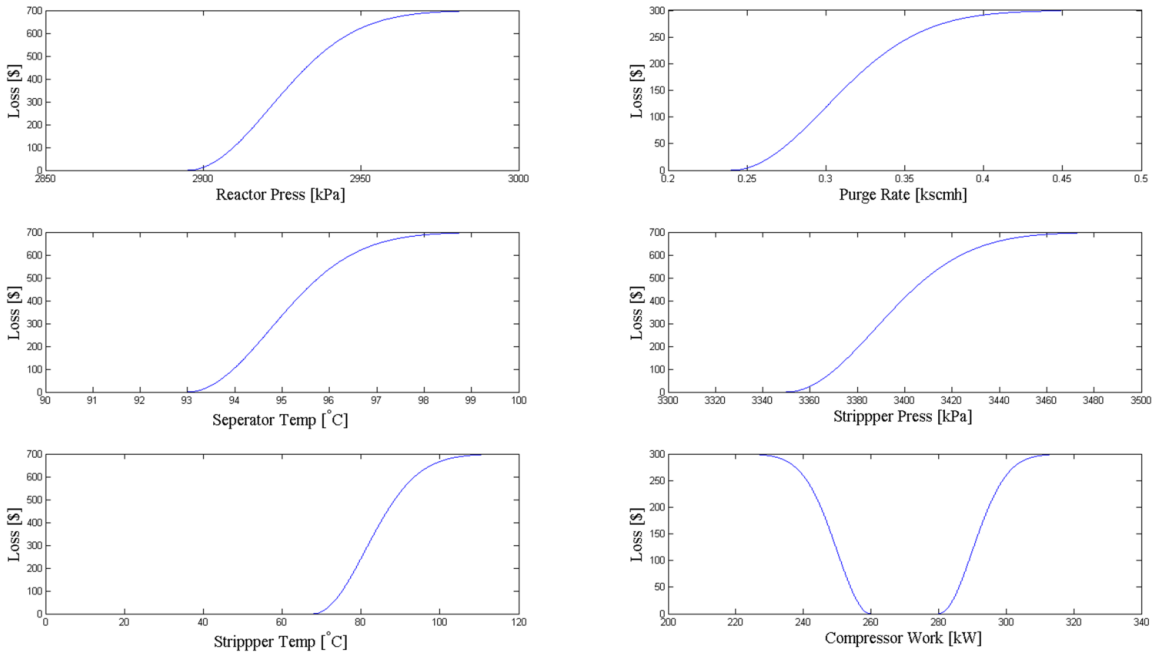

Figure 7-13: Loss functions for IDV13.

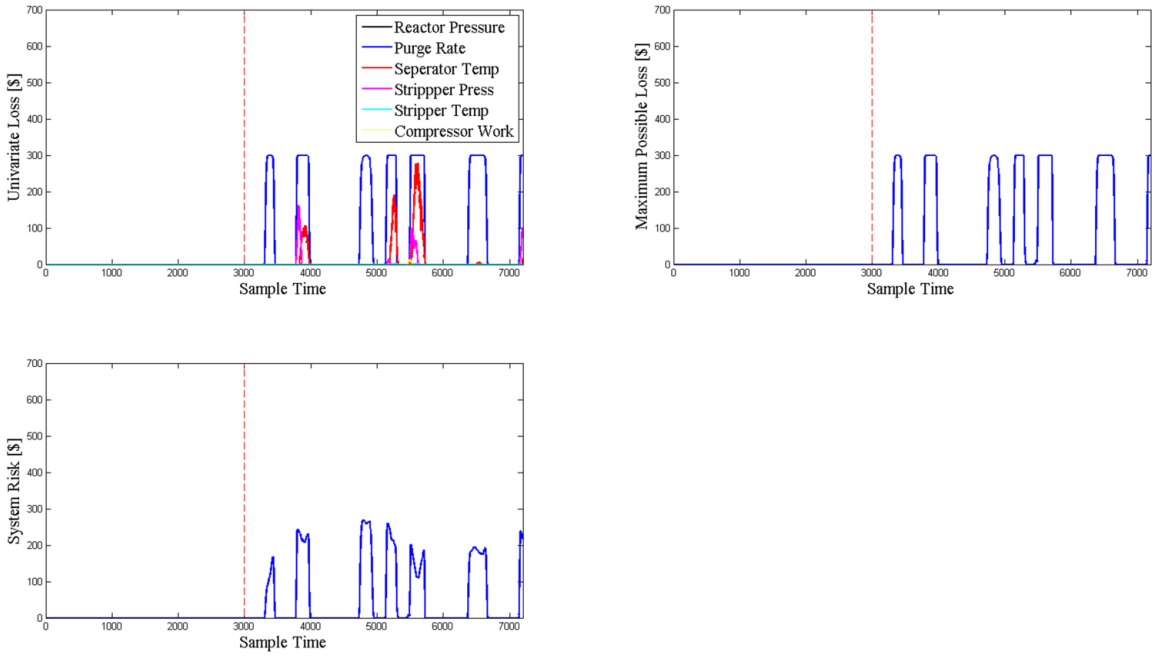

Figure 7-14: Operational risk of TEP under IDV13.

7.5 Conclusions

A risk-based process monitoring technique is proposed for complex process systems. This technique first uses self-organizing map to extract nonlinear features from the process data. The probability density of the process data is also estimated by using a histogram approximation. The advantage of the histogram approximation lies in the fact that it does not make any assumption of the data distribution; it is completely self-organized by the neurons thus representing a more realistic situation. During online

monitoring, the online process data is mapped on the SOM to form a 2D trajectory which can be easily interpreted to identify the operation states of the process. This dynamic trajectory also gives a relative measure of the magnitude of the fault. The deviation of the trajectory is then used to compute the dynamic contribution of each process variable. Those process variables that have high contribution to the propagation of fault are identified. Each of the identified process variables is assigned a loss function to quantify the univariate loss. The loss function used in this study is the inverted normal loss function due to its flexibility to incorporate multiple operational constraints. The maximum possible loss of the process is determined in real-time by taking the maximum univariate loss at each sample interval. Finally, the operational risk of the process is the product of the maximum possible loss and the probability of fault occurring. The proposed technique is tested with the benchmark TEP process. It is demonstrated that the proposed technique provides more robust assessment of the operational risk of process operation. This allows the safety measure or remedial actions to directly target at both the origin and the propagation path of the fault. With this technique implemented, it is possible to bring the process operation back in normal region with minimum cost and delay.

8 Conclusions

For past years, the development of statistics-based process monitoring techniques has revolved around subspace models. It is expected that this direction will continue to follow this paradigm for many years to come. The performance of these subspace models for process monitoring are dependent on three major factors, which have been discussed extensively in this thesis. The first major factor is the assumption of the probability distribution of the latent variables in the feature space. Many subspace models assume that latent variables follow a Gaussian distribution, such that the computation is efficient and tractable. It has been shown in this thesis that this assumption is in fact legitimate under the central limit theorem, particularly for large scale systems, as the latent variables are simply the weighted combination of a large number of process variables. For relatively small-scale systems, extra caution needs to be taken when working with Gaussian distributions. In this respect, a number of effective techniques based on Copula model and Self-Organizing map have been proposed to deal with process variables having dominant non-Gaussian variations.

The second major factor governing the performance of the subspace models is their ability to retain nonlinear correlation structures among process variables. The use of kernel tricks to linearize relationships between process variables in a very high dimensional space has been proven to be a validated approach. In fact, this approach has become a major direction of nonlinear process monitoring. Despite having many virtues, kernel methods suffer from uncertainty in the choice of kernels and hyper parameters, as well as high computational burden. This thesis addressed these drawbacks of the kernel methods by proposing several alternative techniques, including semi-parametric PCA adopting nonlinear correlation measures and nonlinear Gaussian belief network. As compared to kernel methods, all of these proposed techniques have been demonstrated to offer superior process monitoring performance at a lower computational cost.

Lastly, there are many reasons that could lead to contamination of process data for model training. For example, the process data collected at the testing run of a newly deployed industrial process may be contaminated by unstable operating conditions. The robustness of the techniques to data contamination is a key element ensuring feasibility of real-time application of the process monitoring techniques to these new processes, which will significantly reduce chance of false diagnosis. The semi-parametric PCA proposed in this thesis adopted robust correlation measures, such as Spearman's correlation measure and Kendall tau's correlation measure to provide highly accurate process monitoring under different levels of data contamination.

Due to limited time and resources, the proposed work did not consider the situation where operation of process systems constantly switches among a range of modes to meet production requirements. Each of the operating modes has drastically different dynamic characteristics, which cannot be accurately modelled into a single subspace model. In future work, a probabilistic mixture of subspace models will be developed to address this problem. There are two main steps involved in building this mixture model. The first step is to construct a subspace model for each of the operating modes. The second step is to

estimate mixing parameters for each of the subspace models in a similar fashion to Gaussian mixture model, such that the posterior probability of assignment of each online data sample can be calculated. Once an online data sample is aligned to a specific mode, its monitoring statistics can be computed for fault detection and diagnosis.

Process monitoring techniques play an important but limited role in improving the safety of process operations. To complete the circle, these techniques need to be integrated into the operational risk assessment framework to aid decision making process, so as to determine the best remedial strategy to bring the process conditions back to normal at minimal cost. The last two chapters of this thesis provided two directions of research along this path. In future development, these two directions will be further exploited and validated to form a complete risk management package for modern industrial processes.

9 Appendices

9.1 Process Flow Diagram of Tennessee Eastman Process

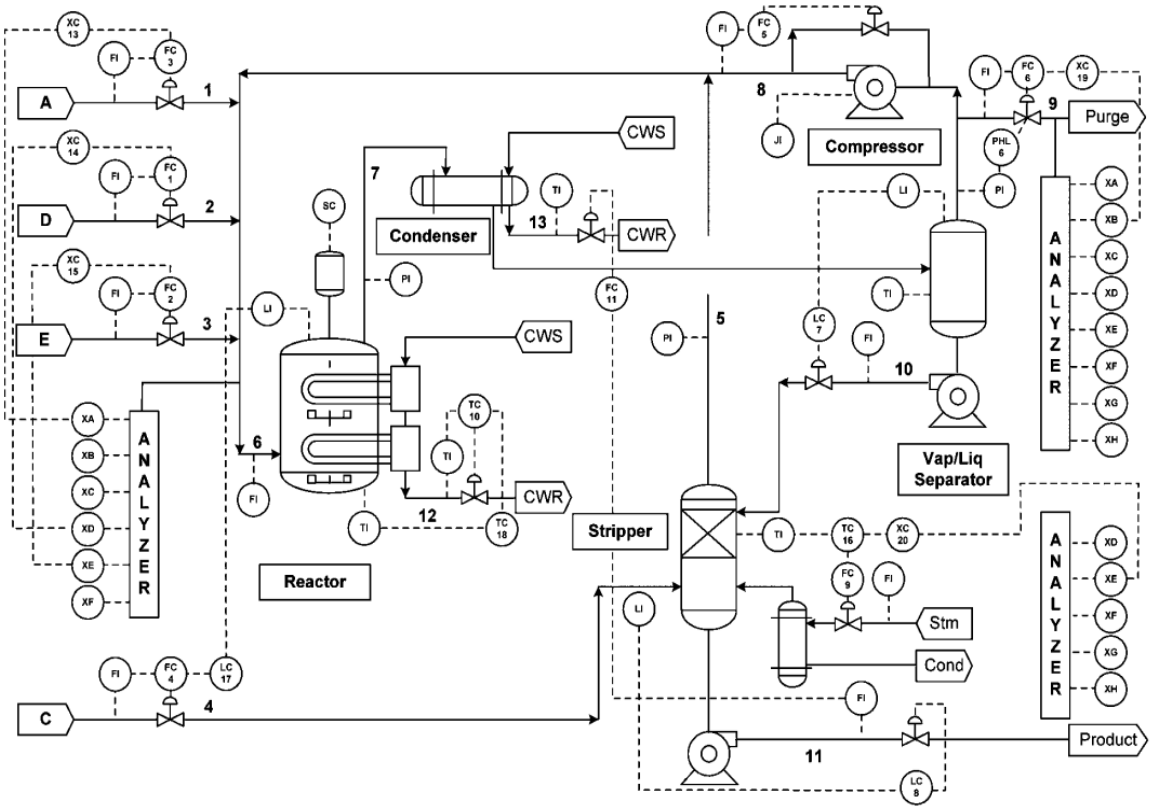

Figure 9-1: Process flow diagram of the Tennessee Eastman chemical process.

9.2 Monitored Process Variables of the TEP

Table 9-1: Monitored variables of the Tennessee Eastman Chemical process.

Variable Number	Variable Description	Unit
Continuously monitored process variables		
X1	A feed (stream 1)	kscmh
X2	D feed (stream 2)	kg/hr
X3	E feed (stream 3)	kg/hr
X4	A and C feed (stream 4)	kscmh
X5	Recycle flow (stream 8)	kscmh
X6	Reactor feed rate (stream 6)	kscmh
X7	Reactor pressure	kPa gauge
X8	Reactor level	%
X9	Reactor temperature	°C
X10	Purge rate (stream 9)	Kscmh
X11	Separator temperature	°C
X12	Separator level	%
X13	Separator pressure	kPa gauge

X14	Separator underflow (stream 10)	m^3/hr
X15	Stripper level	%
X16	Stripper pressure	kPa gauge
X17	Stripper underflow (stream 11)	m^3/hr
X18	Stripper temperature	°C
X19	Stripper steam flow	Kg/hr
X20	Compressor work	kW
X21	Reactor cooling water outlet temperature	°C
X22	Condenser cooling water outlet temperature	°C
	Manipulated process variable	
X23	D feed flow valve (stream 2)	
X24	E feed flow valve (stream 3)	
X25	A feed flow valve (stream 1)	
X26	Total feed flow valve (stream 4)	
X27	Compressor recycle valve	
X28	Purge valve (stream 9)	
X29	Separator pot liquid flow valve (stream 10)	
X30	Stripper liquid product flow valve (stream 11)	
X31	Stripper steam valve	
X32	Reactor cooling water flow	
X33	Condenser cooling water flow	

9.3 Simulated Fault Conditions of the TEP

Table 9-2: Simulated fault conditions of the Tennessee Eastman prcess

Fault No.	Fault description	Signal Type
IDV1	A/C feed ratio, B composition constant (stream 4)	Step
IDV2	B composition, A/C feed ratio constant (stream 4)	Step
IDV3	D feed temp. (stream 2)	Step
IDV4	reactor cooling water inlet temperature	Step
IDV5	condenser cooling water inlet temperature	Step
IDV6	A feed loss (stream 1)	Step
IDV7	C header pressure loss-reduced availability (stream 4)	Step
IDV8	A, B, C feed composition (stream 4)	Random variation
IDV9	D feed temperature (stream 2)	Random variation
IDV10	C feed temperature (stream 4)	Random variation
IDV11	reactor cooling water inlet temperature	Random variation
IDV12	condenser cooling water inlet temperature	Random variation
IDV13	reaction kinetics	Slow drift
IDV14	reactor cooling water valve	Sticking
IDV15	condenser cooling water valve	Sticking

9.4 Proof of Proposition 1

Proof. Let $\mathbf{v}$ be a random unit vector in the subspace spanned by the standardized data matrix $\mathbf{X} = \{\mathbf{x}^i = \{x_1^i, x_2^i, \dots, x_d^i\}\}_{i=1}^N$. The variance of the data samples in the direction of $\mathbf{v}$ is computed as:

$$\hat{\sigma}_{\mathbf{v}} = \frac{1}{N-1}\sum_{i=1}^{N}\left(\left\langle \mathbf{x}^i, \mathbf{v}\right\rangle\right)^2 = \frac{\mathbf{v}^T\mathbf{X}^T\mathbf{X}\mathbf{v}}{N-1} = \mathbf{v}^T\boldsymbol{\Sigma}^0\mathbf{v} \tag{9.1}$$

The correlation matrix in Eq.(9.1) is replaced with its spectral decomposition $\boldsymbol{\Sigma}^0 = \sum_{j=1}^{d}\lambda_j\mathbf{v}_j\mathbf{v}_j^T$. Then Eq. (9.1) is rewritten as following:

$$\hat{\sigma}_{\mathbf{v}} = \sum_{j=1}^{d}\lambda_j(\mathbf{v}_j^T\mathbf{v})^T(\mathbf{v}_j^T\mathbf{v}) \le \lambda_1\sum_{j=1}^{d}(\mathbf{v}_j^T\mathbf{v})^T(\mathbf{v}_j^T\mathbf{v}) \tag{9.2}$$

It is easy see that Eq. (9.2) is maximized $\hat{\sigma}_{\mathbf{v}} = \lambda_1$ when $\mathbf{v} = \mathbf{v}_1$. This means that $\mathbf{v}_1$ lies in the direction of maximum variance. The variance in the direction of $\mathbf{v} = \mathbf{v}_1$ is the largest eigenvalue λ_1.

9.5 Proof of Proposition 2

Proof. First, multiply both sides of Eq. (3.3) by $\mathbf{v}_t$ ($\mathbf{v}_t^T\mathbf{v}_t = ||\mathbf{v}_t||_2 = 1$):

$$\boldsymbol{\Sigma}_t^0\mathbf{v}_t = \boldsymbol{\Sigma}_{t-1}^0\mathbf{v}_t - \mathbf{v}_t\mathbf{v}_t^T\boldsymbol{\Sigma}_{t-1}^0\mathbf{v}_t\mathbf{v}_t^T\mathbf{v}_t = \lambda_t\mathbf{v}_t - \lambda_t\mathbf{v}_t = 0. \tag{9.3}$$

The corresponding eigenvalue of $\mathbf{v}_t$ in the newly formed $\boldsymbol{\Sigma}_t^0$ is set to zero. Suppose $\mathbf{v}_j$ is an arbitrary eigenvector whose associated variance is not removed from $\boldsymbol{\Sigma}_{t-1}^0$. In addition, the eigenvectors of a symmetric correlation matrix are orthonormal $\mathbf{v}_t^T\mathbf{v}_j = 0$, then the following equality holds.

$$\boldsymbol{\Sigma}_t^0\mathbf{v}_j = \boldsymbol{\Sigma}_{t-1}^0\mathbf{v}_j - \mathbf{v}_t\mathbf{v}_t^T\boldsymbol{\Sigma}_{t-1}^0\mathbf{v}_t\mathbf{v}_t^T\mathbf{v}_j = \lambda_j\mathbf{v}_j - 0 = \lambda_j\mathbf{v}_j. \tag{9.4}$$

In conjunction with Proposition 1, it is shown that the deflation matrix removes the variance associated with $\mathbf{v}_t$ from $\boldsymbol{\Sigma}_{t-1}^0$ while preserves the variances associated with the other eigenvectors.

9.6 Proof of Proposition 3

Proof. At step $t = 2$,

$$\boldsymbol{\Sigma}_1^0 = \boldsymbol{\Sigma}^0 - \mathbf{v}_1\mathbf{v}_1^T\boldsymbol{\Sigma}^0\mathbf{v}_1\mathbf{v}_1^T. \tag{9.5}$$

Suppose $\mathbf{z} \in \text{span}\{\mathbf{X}\}$ is an arbitrary unit vector. The variance associated with $\mathbf{z}$ with respect to $\boldsymbol{\Sigma}_1^0$ is given as:

$$\begin{aligned}\hat{\sigma}_{\mathbf{z}} &= \mathbf{z}^T\boldsymbol{\Sigma}_1^0\mathbf{z} = \mathbf{z}^T\left(\boldsymbol{\Sigma}^0 - \mathbf{v}_1\mathbf{v}_1^T\boldsymbol{\Sigma}^0\mathbf{v}_1\mathbf{v}_1^T\right)\mathbf{z} \\ &= \mathbf{z}^T\boldsymbol{\Sigma}^0\mathbf{z} - \mathbf{z}^T\mathbf{v}_1\mathbf{v}_1^T\boldsymbol{\Sigma}^0\mathbf{v}_1\mathbf{v}_1^T\mathbf{z} \\ &= \mathbf{z}^T\boldsymbol{\Sigma}^0\mathbf{z} - \left(\mathbf{v}_1^T\mathbf{z}\right)^T\mathbf{v}_1^T\boldsymbol{\Sigma}^0\mathbf{v}_1\left(\mathbf{v}_1^T\mathbf{z}\right) \\ &= \mathbf{z}^T\boldsymbol{\Sigma}^0\mathbf{z} - \langle\mathbf{v}_1,\mathbf{z}\rangle\mathbf{v}_1^T\boldsymbol{\Sigma}^0\mathbf{v}_1\langle\mathbf{v}_1,\mathbf{z}\rangle.\end{aligned} \tag{9.6}$$

Since $\langle\mathbf{v_1},\mathbf{z}\rangle = ||\mathbf{v}_1||_2||\mathbf{z}||_2\cos\theta = \cos\theta$, where θ is the angle between $\mathbf{v}_1$ and $\mathbf{z}$, Eq. (9.6) is reorganized as:

$$\hat{\sigma}_{\mathbf{z}} = \mathbf{z}^T\boldsymbol{\Sigma}^0\mathbf{z} - \mathbf{v}_1^T\boldsymbol{\Sigma}^0\mathbf{v}_1\cos^2\theta \tag{9.7}$$

As $\boldsymbol{\Sigma}^0 \succcurlyeq 0, \mathbf{z}^T\boldsymbol{\Sigma}^0\mathbf{z} \geq 0$. In addition, $\mathbf{v}_1^T\boldsymbol{\Sigma}^0\mathbf{v_1}\cos^2\theta = \hat{\sigma}_{\mathbf{v_1}}\cos^2\theta \geq 0$. To maximize Eq.(9.7), $\hat{\sigma}_{\mathbf{v_1}}\cos^2\theta$ has to be the first maximized at $\theta = \frac{\pi}{2}$, implying $\mathbf{z}$ is orthonormal to $\mathbf{v_1}$. Then $\mathbf{z}^T\boldsymbol{\Sigma}^0\mathbf{z}$ is maximized under the constraint $\mathbf{z} \perp \mathbf{v_1}$. Based on Eq. (9.2),

$$\hat{\sigma}_{\mathbf{z}} = \sum_{j=1}^{d}\lambda_j(\mathbf{v}_j^T\mathbf{z})^T(\mathbf{v}_j^T\mathbf{z}) = \sum_{j=2}^{d}\lambda_j(\mathbf{v}_j^T\mathbf{z})^T(\mathbf{v}_j^T\mathbf{z}) \leq \lambda_2\sum_{j=2}^{d}(\mathbf{v}_j^T\mathbf{z})^T(\mathbf{v}_j^T\mathbf{z}) \tag{9.8}$$

It is evident that z has to be the eigenvector corresponding to the second largest eigenvalue of $\boldsymbol{\Sigma}^0$ to maximize Eq. (9.7). Subsequently, it is easy to show that, by the method of induction, at step t of the Hotelling's deflation method, the extracted eigenvector $\mathbf{v}_t$ is the t^{th} eigenvector of the initial correlation matrix $\boldsymbol{\Sigma}^0$.

9.7 Proof of a tight bound at maximum

Proof. If $q(\mathbf{y}) = p(\mathbf{y}|\mathbf{x})$, then

$$\begin{aligned}F &= E[\log p(\mathbf{x}\cap\mathbf{y})] - E[\log p(\mathbf{y}\,|\,\mathbf{x})] \\ &= E\left[\log p(\mathbf{x}\cap\mathbf{y}) - \log p(\mathbf{y}\,|\,\mathbf{x})\right] \\ &= E\left[\log p(\mathbf{y}\,|\,\mathbf{x})\,p(\mathbf{x}) - \log p(\mathbf{y}\,|\,\mathbf{x})\right] \\ &= E\left[\log p(\mathbf{y}\,|\,\mathbf{x}) + \log p(\mathbf{x}) - \log p(\mathbf{y}\,|\,\mathbf{x})\right] \\ &= E\left[\log p(\mathbf{x})\right] \\ &= \log p(\mathbf{x})\end{aligned} \tag{9.9}$$

Where, $E[\cdot]$ is the expectation with respect to $q(\mathbf{y})$. Therefore, the bound is proven to be tight.

9.8 Derivatives for sigmoidal and linear functions

For linear functions $f(x) = x$

$$\begin{aligned} M_j(\mu_j,\sigma_j) &= \mu_j \\ V_j(\mu_j,\sigma_j) &= \sigma_j^2 \end{aligned} \tag{9.10}$$

The derivatives with respect to μ_j and $\log\sigma_j^2$ are.

$$\begin{aligned} \frac{\partial M_j(\mu_j,\sigma_j)}{\partial \mu_j} &= 1, \quad \frac{\partial M_j(\mu_j,\sigma_j)}{\partial \log\sigma_j^2} = 0 \\ \frac{\partial V_j(\mu_j,\sigma_j)}{\partial \mu_j} &= 0, \quad \frac{\partial V_j(\mu_j,\sigma_j)}{\partial \log\sigma_j^2} = \sigma_j^2 \end{aligned} \tag{9.11}$$

For sigmoidal functions $f(x) = \frac{1}{1+e^{-x}}$

$$\begin{aligned} M_j(\mu_j,\sigma_j) &= \Phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right) \\ V_j(\mu_j,\sigma_j) &= \Phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right)\left[1-\Phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right)\right]\frac{\sigma_j^2}{\sigma_j^2+\frac{\pi}{2}} \end{aligned} \tag{9.12}$$

The derivatives with respect to μ_j and $\log\sigma_j^2$ are.

$$\frac{\partial M_j(\mu_j,\sigma_j)}{\partial \mu_j} = \frac{1}{\sqrt{1+\sigma_j^2}}\phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right);$$

$$\frac{\partial M_j(\mu_j,\sigma_j)}{\partial \log\sigma_j^2} = -\frac{\mu_j\sigma_j^2}{2(1+\sigma_j^2)^{\frac{3}{2}}}\phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right);$$

$$\frac{\partial V_j(\mu_j,\sigma_j)}{\partial \mu_j} = \frac{\sigma_j^2}{\left(\sigma_j^2+\frac{\pi}{2}\right)\sqrt{1+\sigma_j^2}}\phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right)\left[1-2\Phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right)\right];$$

$$\frac{\partial V_j(\mu_j,\sigma_j)}{\partial \log\sigma_j^2} = \frac{\sigma_j^2}{\left(\sigma_j^2+\frac{\pi}{2}\right)}\left\{\begin{array}{l}\frac{\sigma_j^2}{\left(\sigma_j^2+\frac{\pi}{2}\right)}\Phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right)\left[1-\Phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right)\right] \\ -\frac{\mu_j\sigma_j^2}{2(1+\sigma_j^2)^{\frac{3}{2}}}\phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right)\left[1-2\Phi\left(\frac{\mu_j}{\sqrt{1+\sigma_j^2}}\right)\right]\end{array}\right\}. \quad (9.13)$$

where Φ and ϕ are the cumulative distribution function and probability density function for a standard normal distribution, respectively.

9.9 Derivations of Eqs. (4.17) and (4.19)

Once μ_j^t and σ_j^t are estimated in the E-step, the modal weights w_{ji} can be obtained by maximizing Eq. (4.13) by setting the derivative of Eq. (4.13) to zero with respect to w_{ji} to for each training sample x_i^t.

$$
\begin{aligned}
\frac{\partial F(t)}{\partial w_{ji}} &= \frac{\partial}{\partial w_{ji}} \left\{ -\sum_{i=1}^{N} \frac{\log\left(2\pi\varepsilon_i^2\right)}{2} - \sum_{i=1}^{N} \frac{\left[x_i^t - \sum_{j \in A_i} w_{ji} \alpha_j^t \right]^2 + \sum_{j \in A_i} w_{ji}^2 \beta_j^t}{2\varepsilon_i^2} \right. \\
&\qquad \left. + \sum_{j \in A_i} \frac{\left(1 + \log 2\pi \left(\sigma_j^t\right)^2 - \frac{\left(\sigma_j^t\right)^2}{\varepsilon_i^2} \right)}{2} \right\} \\
&\rightarrow \frac{\partial}{\partial w_{ji}} \left\{ \left[x_i^t - \sum_{j \in A_i} w_{ji} \alpha_j^t \right]^2 + \sum_{j \in A_i} w_{ji}^2 \beta_j^t \right\} \\
&= \frac{\partial}{\partial w_{ji}} \left\{ \left(x_i^t\right)^2 - 2 x_i^t \sum_{j \in A_i} w_{ji} \alpha_j^t + \sum_{k \in A_i} \sum_{j \in A_i} w_{ji}^2 \alpha_j^t \alpha_k^t + \sum_{j \in A_i} w_{ji}^2 \beta_j^t \right\} \\
&= -2 \sum_{j \in A_i} x_i^t \alpha_j^t + 2 \sum_{k \in A_i} \sum_{j \in A_i} w_{ji} \alpha_j^t \alpha_k^t + 2 \sum_{j \in A_i} w_{ji} \beta_j^t = 0 \\
&\rightarrow \sum_{k \in A_i} \sum_{j \in A_i} w_{ji} \alpha_j^t \alpha_k^t + \sum_{j \in A_i} w_{ji} \beta_j^t = \sum_{j \in A_i} x_i^t \alpha_j^t
\end{aligned}
\tag{9.14}
$$

Eq. (9.14) has to be satisfied for each training sample t . Therefore, the following equality is also true.

$$
\frac{1}{T} \sum_{t=1}^{T} \sum_{k \in A_i} \sum_{j \in A_i} w_{ji} \alpha_j^t \alpha_k^t + \frac{1}{T} \sum_{t=1}^{T} \sum_{j \in A_i} w_{ji} \beta_j^t = \frac{1}{T} \sum_{t=1}^{T} \sum_{j \in A_i} x_i^t \alpha_j^t \tag{9.15}
$$

Eq. (9.15) can also be written with respect to a single $j \in A_i$ which recovers Eq. (4.17)

$$
\frac{1}{T} \sum_{t=1}^{T} \sum_{k \in A_i} w_{ji} \alpha_j^t \alpha_k^t + \frac{1}{T} \sum_{t=1}^{T} w_{ji} \beta_j^t = \frac{1}{T} \sum_{t=1}^{T} x_i^t \alpha_k^t \tag{9.16}
$$

Similarly, Eq. (4.19) can be obtained by setting the derivative of (4.13) with respect to ε_j^2 to zero. Notice that $i, j \in \{1,2, \ldots, N\}$.

$$\frac{\partial F(t)}{\partial \varepsilon_j^2} = \frac{\partial}{\partial \varepsilon_j^2}\left\{ -\sum_{i=1}^{N}\frac{\log\left(2\pi\varepsilon_i^2\right)}{2} - \sum_{i=1}^{N}\frac{\left[x_i^t - \sum_{j\in A_i} w_{ji}\alpha_j^t\right]^2 + \sum_{j\in A_i} w_{ji}^2\beta_j^t}{2\varepsilon_i^2} + \sum_{j\in A_i}\frac{\left(1+\log 2\pi\left(\sigma_j^t\right)^2 - \frac{\left(\sigma_j^t\right)^2}{\varepsilon_j^2}\right)}{2} \right\}$$

$$\rightarrow \frac{\partial}{\partial \varepsilon_j^2}\left\{ -\frac{\log\left(2\pi\varepsilon_j^2\right)}{2} - \frac{\left[x_i^t - \sum_{j\in A_i} w_{ji}\alpha_j^t\right]^2 + \sum_{j\in A_i} w_{ji}^2\beta_j^t}{2\varepsilon_j^2} + \sum_{j\in A_i}\frac{\left(1+\log 2\pi\left(\sigma_j^t\right)^2 - \frac{\left(\sigma_j^t\right)^2}{\varepsilon_j^2}\right)}{2} \right\}$$

$$= -\frac{1}{2\varepsilon_j^2} + \frac{\left[x_i^t - \sum_{j\in A_i} w_{ji}\alpha_j^t\right]^2 + \sum_{j\in A_i} w_{ji}^2\beta_j^t}{2\left(\varepsilon_j^2\right)^2} + \sum_{j\in A_i}\frac{\left(\sigma_j^t\right)^2}{2\left(\varepsilon_j^2\right)^2} = 0$$

$$\rightarrow \frac{1}{2\varepsilon_j^2} = \frac{\left[x_i^t - \sum_{j\in A_i} w_{ji}\alpha_j^t\right]^2 + \sum_{j\in A_i} w_{ji}^2\beta_j^t}{2\left(\varepsilon_j^2\right)^2} + \sum_{j\in A_i}\frac{\left(\sigma_j^t\right)^2}{2\left(\varepsilon_j^2\right)^2}$$

$$\rightarrow \varepsilon_j^2 = \left[x_i^t - \sum_{j\in A_i} w_{ji}\alpha_j^t\right]^2 + \sum_{j\in A_i} w_{ji}^2\beta_j^t + \sum_{j\in A_i}\left(\sigma_j^t\right)^2 \tag{9.17}$$

Eq. (9.17) holds for every single training step *t.* Similar to Eq. (9.16), Eq. (9.17) can be rewritten as Eq. (9.18) which is equivalent to Eq. (4.19).

$$\varepsilon_j^2 = \sum_{t=1}^{T}\left[x_i^t - \sum_{j\in A_i} w_{ji}\alpha_j^t\right]^2 + \sum_{t=1}^{T}\sum_{j\in A_i} w_{ji}^2\beta_j^t + \sum_{t=1}^{T}\sum_{j\in A_i}\left(\sigma_j^t\right)^2 \tag{9.18}$$

9.10 Sum-product Algorithm

Considering a simple a simple system with n input variables $x_{1:n}$, 2 intermediate variables x_j and x_k and 1 output variable x_l. A schematic representation of this simple system is shown in Figure 9-2.

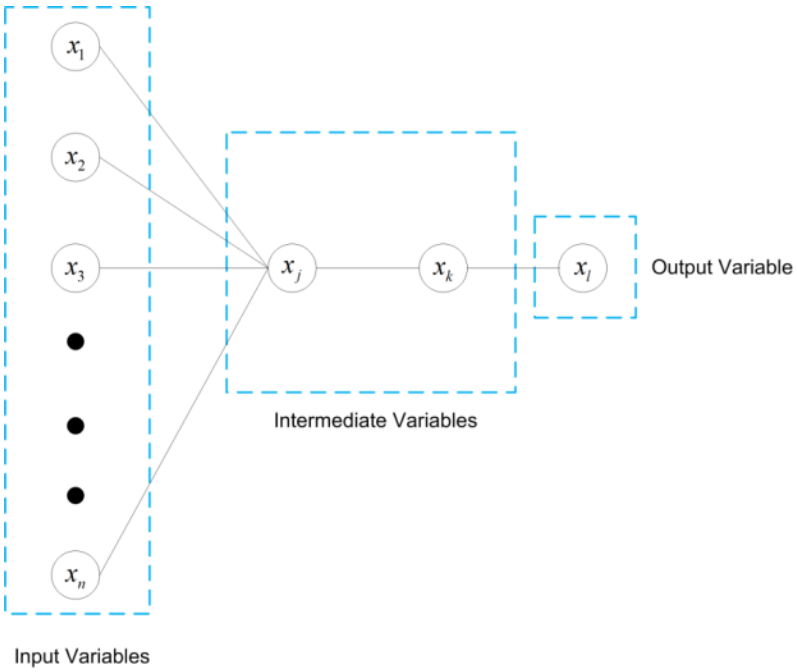

Figure 9-2: Schematic presentation of a simple multivariate system.

The factorized BN is shown below in Figure 9-3. Each variable node has two states, s_0 for fault and s_1 for normal. It is assumed that the output variable x_l has been identified as the faulty monitored variable in the first-stage diagnosis, $P(x_l = s_0) = 1$. The message passed from the ancestor node to factorial node is denoted as $\mu_{x \to f}$. The first inferential task is to calculate the posterior probability of the intermediate variable $P(x_k \mid x_l = s_0)$ through sum product algorithm.

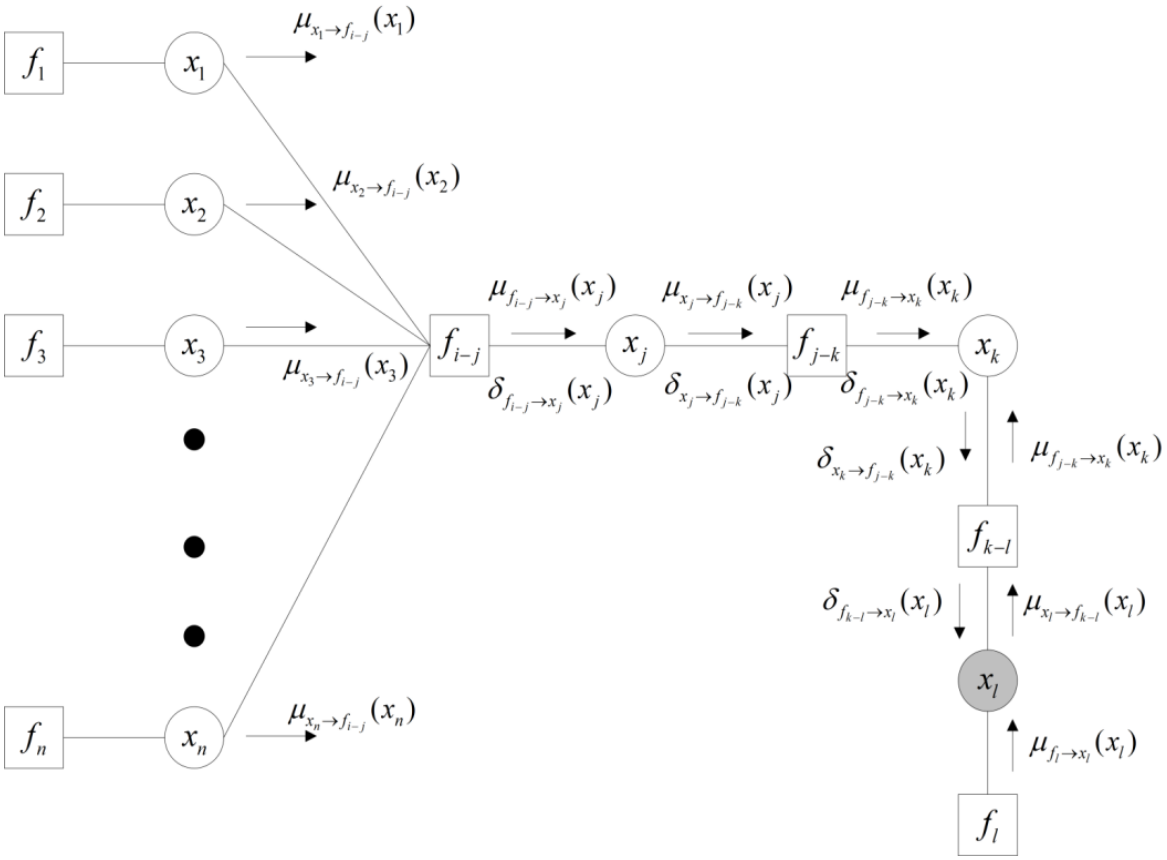

Figure 9-3: Illustrative BN example for sum-product and max-product algorithm.

Sum-product algorithm

1) Message: $f_i \to x_i$

$$\mu_{f_i \to x_i}(x_i) = P(x_i) \tag{9.19}$$

2) Message: $x_{1:n} \to f_i$

$$\mu_{x_i \to f_{i-j}}(x_i) = \mu_{f_i \to x_i}(x_i) = P(x_i) \tag{9.20}$$

3) Message: $f_{i-j} \to x_j$

$$f_{i-j}(x_j, x_{1:n}) = P(x_j \mid x_{1:n})$$

$$\mu_{f_{i-j} \to x_j}(x_j) = \overbrace{\sum_{\{x_{1:n}\}}}^{\text{Sum-out}} f_{i-j}(x_j, x_{1:n}) \overbrace{\prod_{i=1}^{n}}^{\text{Product}} \mu_{x_i \to f_{i-j}}(x_i) \tag{9.21}$$

4) Message: $x_j \to f_{j-k}$

$$\mu_{x_j \to f_{j-k}}(x_j) = \mu_{f_{i-j} \to x_j}(x_j) \tag{9.22}$$

5) Message: $f_{j-k} \to x_k$

$$\begin{aligned} f_{j-k}(x_j, x_k) &= P(x_k \mid x_j) \\ \mu_{f_{j-k} \to x_k}(x_k) &= \sum_{x_j} f_{j-k}(x_j, x_k) \mu_{f_{i-j} \to x_j}(x_j) \end{aligned} \tag{9.23}$$

6) Message: $f_l \to x_l$

$$e \to \begin{cases} \mu_{f_l \to x_l}(x_l = s_0) = P(x_l = s_0) = 1 \\ \mu_{f_l \to x_l}(x_l = s_1) = P(x_l = s_1) = 0 \end{cases} \tag{9.24}$$

7) Message: $x_l \to f_{k-l}$

$$\begin{aligned} \mu_{x_l \to f_{k-l}}(x_l = s_0) &= \mu_{f_l \to x_l}(x_l = s_0) = 1 \\ \mu_{x_l \to f_{k-l}}(x_l = s_1) &= \mu_{f_l \to x_l}(x_l = s_1) = 0 \end{aligned} \tag{9.25}$$

8) Message: $f_{k-l} \to x_k$

$$\begin{aligned} f_{k-l}(x_k, x_l) &= P(x_l \mid x_k) \\ \mu_{f_{k-l} \to x_k}(x_k) &= \sum_{x_l} f_{k-l}(x_k, x_l) \mu_{x_l \to f_{k-l}}(x_l) \end{aligned} \tag{9.26}$$

9) Posterior probability: $P(x_k \mid x_l = s_0)$

$$P(x_k \mid x_l = s_0) = \mu_{f_{j-k} \to x_k}(x_k) \mu_{f_{k-l} \to x_k}(x_k) \tag{9.27}$$

Max-product algorithm

1) Message $f_i \to x_i$ and $x_{1:n} \to f_i$ are the same as the sum-product algorithm

2) Message: $f_{i-j} \to x_j$

$$\delta_{f_{i-j} \to x_j}(x_j) = \overbrace{\max_{x_{1:n}} \left\{ f_{i-j}(x_j, x_{1:n}) \overbrace{\prod_{i=1}^{n} \mu_{x_i \to f_{i-j}}(x_i)}^{\text{Product}} \right\}}^{\text{Max-out}} \tag{9.28}$$

3) Message $x_j \to f_{j-k}$

$$\delta_{x_j \to f_{j-k}}(x_j) = \delta_{f_{i-j} \to x_j}(x_j) \tag{9.29}$$

4) Message: $f_{j-k} \to x_k$

$$\delta_{f_{j-k} \to x_k}(x_k) = \max_{x_j} \left\{ f_{j-k}(x_j, x_k) \delta_{f_{i-j} \to x_j}(x_j) \right\} \tag{9.30}$$

5) Message: $x_k \to f_{k-l}$

$$\delta_{x_k \to f_{k-l}}(x_k) = \delta_{f_{j-k} \to x_k}(x_k) \tag{9.31}$$

6) Message: $f_{k-l} \to x_l$

$$\delta_{f_{k-l} \to x_l}(x_l) = \max_{x_k} \left\{ f_{k-l}(x_k, x_l) \delta_{x_k \to f_{k-l}}(x_k) \right\} \tag{9.32}$$

7) Maximize probability

$$\max_{x_k} P(x_l = s_0) = \max_{x_k} \delta_{f_{k-l} \to x_l}(x_l = \mathrm{s}_0) \tag{9.33}$$

Subsequently, the most likely state of each node is retrieved by back-tracking.

Back-tracking step

1) $x_l \to x_k$

$$x_k^{\max} = \arg\max_{x_k} \left\{ f_{k-l}(x_k, x_l) \delta_{x_k \to f_{k-l}}(x_k) \right\} \tag{9.34}$$

2) $x_k \to x_j$

$$x_j^{\max} = \arg\max_{x_j} \left\{ f_{j-k}(x_j, x_k) \delta_{f_{i-j} \to x_j}(x_j) \right\} \tag{9.35}$$

3) $x_j \to x_{1:n}$

$$\left\{ x_{1:n}^{\max} \right\} = \arg\max_{x_{1:n}} \left\{ f_{i-j}(x_j, x_{1:\mathrm{n}}) \prod_{i=1}^{n} \mu_{x_i \to f_{i-j}}(x_i) \right\} \tag{9.36}$$

The set of nodes most likely in faulty state is selected as the ones satisfying the following conditions

$$\begin{aligned} & x_m \in \{x_k^{\max}, x_j^{\max}, x_{1:n}^{\max}\} \\ & x_m = s_0 \end{aligned} \tag{9.37}$$

Finally, the true root-cause variable is identified as

$$x_{true-root} = \arg\max_{x_m} P(x_m = s_0) \tag{9.38}$$

where $P(x_m = s_0)$ is the posterior probability of x_m determined by the sum-product algorithm.

10 Bibliography

1. Lee J-M, Yoo C, Lee I-B. Statistical process monitoring with independent component analysis. *J. Process Control.* 2004;14(5):467-485.

2. Kano M, Nagao K, Hasebe S, et al. Comparison of multivariate statistical process monitoring methods with applications to the Eastman challenge problem. *Comput. Chem. Eng.* 2002;26(2):161-174.

3. Kresta JV, MacGregor JF, Marlin TE. Multivariate statistical monitoring of process operating performance. *Can. J. Chem. Eng.* 1991;69(1):35-47.

4. Bersimis S, Psarakis S, Panaretos J. Multivariate statistical process control charts: an overview. *Qual. Reliab. Eng. Int.* 2007;23(5):517-543.

5. Cherry GA, Qin SJ. Multiblock principal component analysis based on a combined index for semiconductor fault detection and diagnosis. *Semiconductor Manufacturing, IEEE Transactions on.* 2006;19(2):159-172.

6. Chiang LH, Braatz RD, Russell EL. *Fault detection and diagnosis in industrial systems*. 1 ed. London: Springer-Verlag; 2001.

7. Bakshi BR. Multiscale PCA with application to multivariate statistical process monitoring. *AIChE J.* 1998;44(7):1596-1610.

8. Kano M, Hasebe S, Hashimoto I, Ohno H. A new multivariate statistical process monitoring method using principal component analysis. *Comput. Chem. Eng.* 2001;25(7):1103-1113.

9. AlGhazzawi A, Lennox B. Monitoring a complex refining process using multivariate statistics. *Control Eng. Pract.* 2008;16(3):294-307.

10. Gertler J. *Fault detection and diagnosis in engineering systems*: CRC press; 1998.

11. Hinton GE, Ghahramani Z. Generative models for discovering sparse distributed representations. *Philosophical Transactions of the Royal Society of London. Series B: Biological Sciences.* 1997;352(1358):1177-1190.

12. Yu J. A nonlinear kernel Gaussian mixture model based inferential monitoring approach for fault detection and diagnosis of chemical processes. *Chem. Eng. Sci.* 2012;68(1):506-519.

13. Piovoso MJ, Hoo K. Multivariate statistics for process control. *IEEE Control Systems Magazine.* 2002;22(5):8-9.

14. Hwang D-H, Han C. Real-time monitoring for a process with multiple operating modes. *Control Eng. Pract.* 1999;7(7):891-902.

15. Lee JM, Qin SJ, Lee IB. Fault detection and diagnosis based on modified independent component analysis. *AIChE J.* 2006;52(10):3501-3514.

16. Hyvärinen A, Oja E. Independent component analysis: algorithms and applications. *Neural Networks.* 2000;13(4):411-430.

17. Hyvarinen A. Fast and robust fixed-point algorithms for independent component analysis. *Neural Networks, IEEE Transactions on.* 1999;10(3):626-634.

18. Lee JM, Qin SJ, Lee IB. Fault detection of non-linear processes using kernel independent component analysis. *The Canadian Journal of Chemical Engineering.* 2007;85(4):526-536.

19. Abu-Mostafa YS, Magdon-Ismail M, Lin H-T. *Learning from data*: AMLBook; 2012.

20. Lee J-M, Yoo C, Choi SW, Vanrolleghem PA, Lee I-B. Nonlinear process monitoring using kernel principal component analysis. *Chem. Eng. Sci.* 2004;59(1):223-234.

21. Rüschendorf L. On the distributional transform, Sklar's theorem, and the empirical copula process. *Journal of Statistical Planning and Inference.* 2009;139(11):3921-3927.

22. Cherubini U, Luciano E, Vecchiato W. *Copula methods in finance*: John Wiley & Sons; 2004.

23. Sklar A. Random variables, distribution functions, and copulas: a personal look backward and forward. *Lecture notes-monograph series.* 1996:1-14.

24. Sall J, Lehman A, Stephens ML, Creighton L. *JMP start statistics: a guide to statistics and data analysis using JMP*: SAS Institute; 2012.

25. Hoff PD. Extending the rank likelihood for semiparametric copula estimation. *The Annals of Applied Statistics.* 2007:265-283.

26. Mohseni Ahooyi T, Arbogast JE, Soroush M. Rolling Pin Method: Efficient General Method of Joint Probability Modeling. *Ind. Eng. Chem. Res.* 2014;53(52):20191-20203.

27. Rüschendorf L. Mathematical risk analysis. *Springer Ser. Oper. Res. Financ. Eng. Springer, Heidelberg.* 2013.

28. Tanizaki H. *Computational methods in statistics and econometrics*: CRC Press; 2004.

29. Demarta S, McNeil AJ. The t copula and related copulas. *International statistical review.* 2005;73(1):111-129.

30. Boyd S, Vandenberghe L. *Convex optimization*: Cambridge university press; 2004.

31. Scheinberg K. An efficient implementation of an active set method for SVMs. *The Journal of Machine Learning Research.* 2006;7:2237-2257.

32. Downs JJ, Vogel EF. A plant-wide industrial process control problem. *Comput. Chem. Eng.* 1993;17(3):245-255.

33. Joe Qin S. Statistical process monitoring: basics and beyond. *J. Chemom.* 2003;17(8-9):480-502.

34. Gnanadesikan R. *Methods for statistical data analysis of multivariate observations.* Vol 321. New York: John Wiley & Sons; 2011.

35. Hair JF, Black WC, Babin BJ, Anderson RE, Tatham RL. *Multivariate data analysis.* Vol 6: Pearson Prentice Hall Upper Saddle River, NJ; 2006.

36. Roweis S. EM algorithms for PCA and SPCA. *Advances in neural information processing systems.* 1998:626-632.

37. Borgognone MaG, Bussi J, Hough G. Principal component analysis in sensory analysis: covariance or correlation matrix? *Food quality and preference.* 2001;12(5):323-326.

38. Han F, Liu H. High dimensional semiparametric scale-invariant principal component analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on.* 2014;36(10):2016-2032.

39. De la Torre F, Black MJ. Robust principal component analysis for computer vision. Paper presented at: Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on2001.

40. Johnson RA, Wichern DW. *Applied multivariate statistical analysis.* Vol 4: Prentice hall Englewood Cliffs, NJ; 1992.

41. Hyvärinen A, Karhunen J, Oja E. *Independent component analysis.* Vol 46. New York: John Wiley & Sons; 2004.

42. Yu H, Khan F, Garaniya V. Modified Independent Component Analysis and Bayesian Network-Based Two-Stage Fault Diagnosis of Process Operations. *Ind. Eng. Chem. Res.* 2015;54(10):2724-2742.

43. Lee T-W, Lewicki MS, Sejnowski TJ. ICA mixture models for unsupervised classification of non-Gaussian classes and automatic context switching in blind signal separation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on.* 2000;22(10):1078-1089.

44. Shimizu S, Hoyer PO, Hyvärinen A, Kerminen A. A linear non-Gaussian acyclic model for causal discovery. *The Journal of Machine Learning Research.* 2006;7:2003-2030.

45. Rice J. *Mathematical statistics and data analysis*. 3 ed. Belmont: Cengage Learning; 2006.

46. Boudt K, Cornelissen J, Croux C. The Gaussian rank correlation estimator: robustness properties. *Statistics and Computing.* 2012;22(2):471-483.

47. Zar JH. Spearman rank correlation. *Encyclopedia of Biostatistics.* 1998.

48. Mackey LW. Deflation methods for sparse PCA. Paper presented at: Advances in neural information processing systems2009.

49. Zhang Z, Zha H, Simon H. Low-rank approximations with sparse factors I: Basic algorithms and error analysis. *SIAM Journal on Matrix Analysis and Applications.* 2002;23(3):706-727.

50. Zhang Z, Zha H, Simon H. Low-rank approximations with sparse factors II: Penalized methods with discrete Newton-like iterations. *SIAM journal on matrix analysis and applications.* 2004;25(4):901-920.

51. Croux C, Dehon C. Influence functions of the Spearman and Kendall correlation measures. *Statistical methods & applications.* 2010;19(4):497-515.

52. Foster DV, Grassberger P. Lower bounds on mutual information. *Physical Review E.* 2011;83(1):010101.

53. Granger C, Lin J-L. Using the mutual information coefficient to identify lags in nonlinear models. *Journal of time series analysis.* 1994;15(4):371-384.

54. Saad Y. Projection and deflation method for partial pole assignment in linear state feedback. *Automatic Control, IEEE Transactions on.* 1988;33(3):290-297.

55. JH W, FL B, C R. *Linear Algebra.* Vol 2. 1 ed. Berlin, Heidelberg: Springer-Verlag; 2013.

56. d'Aspremont A, El Ghaoui L, Jordan MI, Lanckriet GR. A direct formulation for sparse PCA using semidefinite programming. *SIAM review.* 2007;49(3):434-448.

57. Moghaddam B, Weiss Y, Avidan S. Spectral bounds for sparse PCA: Exact and greedy algorithms. Paper presented at: Advances in neural information processing systems2005.

58. Sriperumbudur BK, Torres DA, Lanckriet GR. Sparse eigen methods by dc programming. Paper presented at: Proceedings of the 24th international conference on Machine learning2007.

59. Moghaddam B, Weiss Y, Avidan S. Generalized spectral bounds for sparse LDA. Paper presented at: Proceedings of the 23rd international conference on Machine learning2006.

60. Grant M, Boyd S. CVX: Matlab software for disciplined convex programming (web page and software). *URL http://stanford. edu/boyd/cvx.* 2008.

61. Candès EJ, Romberg J, Tao T. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *Information Theory, IEEE Transactions on.* 2006;52(2):489-509.

62. Botev Z, Grotowski J, Kroese D. Kernel density estimation via diffusion. *The Annals of Statistics.* 2010;38(5):2916-2957.

63. Thornhill NF, Patwardhan SC, Shah SL. A continuous stirred tank heater simulation model with applications. *J. Process Control.* 2008;18(3):347-360.

64. Wu W, Massart D, De Jong S. The kernel PCA algorithms for wide data. Part I: theory and algorithms. *Chemom. Intell. Lab. Syst.* 1997;36(2):165-172.

65. Venkatasubramanian V, Rengaswamy R, Yin K, Kavuri SN. A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Comput. Chem. Eng.* 2003;27(3):293-311.

66. Var I. Multivariate data analysis. *vectors.* 1998;8(2):125-136.

67. MacGregor JF, Jaeckle C, Kiparissides C, Koutoudi M. Process monitoring and diagnosis by multiblock PLS methods. *AIChE J.* 1994;40(5):826-838.

68. Jolliffe I. *Principal component analysis*: Wiley Online Library; 2005.

69. Cao L, Chua K, Chong W, Lee H, Gu Q. A comparison of PCA, KPCA and ICA for dimensionality reduction in support vector machine. *Neurocomputing.* 2003;55(1):321-336.

70. Chen J, Bandoni JA, Romagnoli JA. Robust PCA and normal region in multivariate statistical process monitoring. *AIChE J.* 1996;42(12):3563-3566.

71. Xu H, Caramanis C, Sanghavi S. Robust PCA via outlier pursuit. Paper presented at: Advances in Neural Information Processing Systems2010.

72. Martin E, Morris A. Non-parametric confidence bounds for process performance monitoring charts. *J. Process Control.* 1996;6(6):349-358.

73. Hyvärinen A, Oja E. A fast fixed-point algorithm for independent component analysis. *Neural Comput.* 1997;9(7):1483-1492.

74. Meinecke FC, Harmeling S, Müller K-R. Robust ICA for super-gaussian sources. *Independent Component Analysis and Blind Signal Separation*: Springer; 2004:217-224.

75. Pandey S, Billor N, Turkmen A. The effect of outliers in independent component analysis. *American Journal of Mathematical and Management Sciences.* 2008;28(3-4):399-418.

76. Wang J, He QP. Multivariate statistical process monitoring based on statistics pattern analysis. *Ind. Eng. Chem. Res.* 2010;49(17):7858-7869.

77. Kocsor A, Csirik J. Fast independent component analysis in kernel feature spaces. Paper presented at: SOFSEM 2001: Theory and Practice of Informatics2001.

78. Bach FR, Jordan MI. Kernel independent component analysis. *The Journal of Machine Learning Research.* 2003;3:1-48.

79. Kocsor A, Tóth L. Kernel-based feature extraction with a speech technology application. *Signal Processing, IEEE Transactions on.* 2004;52(8):2250-2263.

80. Yang J, Gao X, Zhang D, Yang J-y. Kernel ICA: An alternative formulation and its application to face recognition. *Pattern Recognition.* 2005;38(10):1784-1787.

81. Liu X, Kruger U, Littler T, Xie L, Wang S. Moving window kernel PCA for adaptive monitoring of nonlinear processes. *Chemom. Intell. Lab. Syst.* 2009;96(2):132-143.

82. Frey BJ, Hinton GE. Variational learning in nonlinear Gaussian belief networks. *Neural Comput.* 1999;11(1):193-213.

83. Neal RM, Hinton GE. A view of the EM algorithm that justifies incremental, sparse, and other variants. *Learning in graphical models*. 1 ed. Netherlands: Springer; 1998:355-368.

84. Murphy KP. *Machine learning: a probabilistic perspective*: MIT Press; 2012.

85. Chin T-J, Suter D. Incremental kernel principal component analysis. *Image Processing, IEEE Transactions on.* 2007;16(6):1662-1674.

86. Lawrence Ricker N. Decentralized control of the Tennessee Eastman challenge process. *J. Process Control.* 1996;6(4):205-221.

87. Kano M, Hasebe S, Hashimoto I, Ohno H. Statistical process monitoring based on dissimilarity of process data. *AIChE J.* 2002;48(6):1231-1240.

88. Russell EL, Chiang LH, Braatz RD. Fault detection in industrial processes using canonical variate analysis and dynamic principal component analysis. *Chemom. Intell. Lab. Syst.* 2000;51(1):81-93.

89. Venkatasubramanian V, Rengaswamy R, Kavuri SN, Yin K. A review of process fault detection and diagnosis: Part III: Process history based methods. *Comput. Chem. Eng.* 2003;27(3):327-346.

90. Kosanovich KA, Dahl KS, Piovoso MJ. Improved process understanding using multiway principal component analysis. *Ind. Eng. Chem. Res.* 1996;35(1):138-146.

91. Abdi H, Williams LJ. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics.* 2010;2(4):433-459.

92. Lee J-M, Yoo C, Lee I-B. Statistical monitoring of dynamic processes based on dynamic independent component analysis. *Chem. Eng. Sci.* 2004;59(14):2995-3006.

93. Ge Z, Song Z. Process monitoring based on independent component analysis-principal component analysis (ICA-PCA) and similarity factors. *Ind. Eng. Chem. Res.* 2007;46(7):2054-2063.

94. Paninski L. Estimation of entropy and mutual information. *Neural Comput.* 2003;15(6):1191-1253.

95. MacKay DJ. *Information theory, inference and learning algorithms*: Cambridge university press; 2003.

96. Torkkola K. Feature extraction by non parametric mutual information maximization. *The Journal of Machine Learning Research.* 2003;3:1415-1438.

97. Yu J, Rashid MM. A novel dynamic bayesian network-based networked process monitoring approach for fault detection, propagation identification, and root cause diagnosis. *AIChE J.* 2013;59(7):2348-2365.

98. Darwiche A. *Modeling and reasoning with Bayesian networks*: Cambridge University Press; 2009.

99. Bobbio A, Portinale L, Minichino M, Ciancamerla E. Improving the analysis of dependable systems by mapping fault trees into Bayesian networks. *Reliability Engineering & System Safety.* 2001;71(3):249-260.

100. Cichocki A, Amari S-i. *Adaptive blind signal and image processing*: John Wiley Chichester; 2002.

101. Jiang Q, Yan X. Joint probability density and double-weighted independent component analysis for multimode non-Gaussian process monitoring. *Ind. Eng. Chem. Res.* 2014.

102. Rashid MM, Yu J. A new dissimilarity method integrating multidimensional mutual information and independent component analysis for non-Gaussian dynamic process monitoring. *Chemom. Intell. Lab. Syst.* 2012;115:44-58.

103. Chen J, Yu J. Independent Component Analysis Mixture Model Based Dissimilarity Method for Performance Monitoring of Non-Gaussian Dynamic Processes with Shifting Operating Conditions. *Ind. Eng. Chem. Res.* 2013;53(13):5055-5066.

104. Yuan Z, Khakzad N, Khan F, Amyotte P. Risk analysis of dust explosion scenarios using Bayesian networks. *Risk Anal.* 2014.

105. Cai B, Liu Y, Liu Z, Tian X, Dong X, Yu S. Using Bayesian networks in reliability evaluation for subsea blowout preventer control system. *Reliability Engineering & System Safety.* 2012;108:32-41.

106. Jensen FV. *An introduction to Bayesian networks.* Vol 210: UCL press London; 1996.

107. Nielsen TD, Jensen FV. *Bayesian networks and decision graphs*: Springer; 2009.

108. Murphy KP. *Dynamic bayesian networks: representation, inference and learning*, University of California, Berkeley; 2002.

109. Yu H, Khan F, Garaniya V, Ahmad A. Self-organizing map based fault diagnosis technique for non-Gaussian processes. *Ind. Eng. Chem. Res.* 2014.

110. Westerhuis JA, Gurden SP, Smilde AK. Generalized contribution plots in multivariate statistical process monitoring. *Chemom. Intell. Lab. Syst.* 2000;51(1):95-114.

111. Kschischang FR, Frey BJ, Loeliger H-A. Factor graphs and the sum-product algorithm. *Information Theory, IEEE Transactions on.* 2001;47(2):498-519.

112. Loeliger H-A. An introduction to factor graphs. *Signal Processing Magazine, IEEE.* 2004;21(1):28-41.

113. Koller D, Friedman N. *Probabilistic graphical models: principles and techniques*: MIT press; 2009.

114. Druzdzel MJ. SMILE: Structural Modeling, Inference, and Learning Engine and GeNIe: a development environment for graphical decision-theoretic models. Paper presented at: AAAI/IAAI1999.

115. Dash S, Venkatasubramanian V. Challenges in the industrial applications of fault diagnostic systems. *Comput. Chem. Eng.* 2000;24(2):785-791.

116. Kohonen T, Oja E, Simula O, Visa A, Kangas J. Engineering applications of the self-organizing map. *Proceedings of the IEEE.* 1996;84(10):1358-1384.

117. Chopra T, Vajpai J. Classification of Faults in DAMADICS Benchmark Process Control System Using Self Organizing Maps. *International Journal of Soft Computing.*1.

118. Gonçalves LF, Bosa JL, Balen TR, Lubaszewski MS, Schneider EL, Henriques RV. Fault detection, diagnosis and prediction in electrical valves using self-organizing maps. *Journal of Electronic Testing.* 2011;27(4):551-564.

119. Zhong F, Shi T, He T. Fault diagnosis of motor bearing using self-organizing maps. Paper presented at: Electrical Machines and Systems, 2005. ICEMS 2005. Proceedings of the Eighth International Conference on2005.

120. Vapola M, Simula O, Kohonen T, Meriläinen P. Representation and identification of fault conditions of an anaesthesia system by means of the Self-Organizing Map. *ICANN'94*: Springer; 1994:350-353.

121. Gonçalves LF, Schneider EL, Henriques RVB, Lubaszewski M, Bosa JL, Engel PM. Fault prediction in electrical valves using temporal Kohonen maps. Paper presented at: Test Workshop (LATW), 2010 11th Latin American2010.

122. Sirola M, Talonen J, Lampi G. SOM based methods in early fault detection of nuclear industry. Paper presented at: Proceedings of the 17th European Symposium On Artificial Neural Networks, ESANN2009.

123. Vesanto J. SOM-based data visualization methods. *Intelligent data analysis.* 1999;3(2):111-126.

124. Zadakbar O, Imtiaz S, Khan F. Dynamic risk assessment and fault detection using principal component analysis. *Ind. Eng. Chem. Res.* 2012;52(2):809-816.

125. Kohonen T. The self-organizing map. *Proceedings of the IEEE.* 1990;78(9):1464-1480.
126. Kandel ER, Schwartz JH, Jessell TM. *Principles of neural science.* Vol 4: McGraw-Hill New York; 2000.
127. Fodor IK. A survey of dimension reduction techniques: Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory; 2002.
128. Ng Y, Srinivasan R. Monitoring of distillation column operation through self-organizing maps. Paper presented at: Dynamics and Control of Process Systems 2004 (DYCOPS-7): A Proceedings Volume from the 7th IFAC Symposium, Cambridge, Massachusetts, USA, 5-7 July 20042004.
129. Wu S, Chow TW, Huang D. Visualization of Induction Machine Fault Detection Using Self-Organizing Map and Support Vector Machine.
130. Bao H, Khan F, Iqbal T, Chang Y. Risk-based fault diagnosis and safety management for process systems. *Process Saf. Prog.* 2011;30(1):6-17.
131. Ericson CA. *Hazard analysis techniques for system safety*: Wiley-Interscience; 2005.
132. Dunia R, Joe Qin S. Subspace approach to multidimensional fault identification and reconstruction. *AIChE J.* 1998;44(8):1813-1831.
133. Zadakbar O, Khan F, Imtiaz S. Development of economic consequence methodology for process risk analysis. *Risk Anal.* 2014.
134. Hashemi SJ, Ahmed S, Khan F. Loss functions and their applications in process safety assessment. *Process Saf. Prog.* 2014.
135. Hashemi SJ, Ahmed S, Khan FI. Risk-based operational performance analysis using loss functions. *Chem. Eng. Sci.* 2014;116:99-108.
136. Spiring FA. The reflected normal loss function. *Can. J. Stat.* 1993;21(3):321-330.
137. Chang Y, Khan F, Ahmed S. A risk-based approach to design warning system for processing facilities. *Process Saf. Environ. Prot.* 2011;89(5):310-316.
138. Kohonen T. *Self-organizing maps.* Vol 30. 3 ed. Berlin: Springer-Verlag; 2001.
139. Vesanto J, Himberg J, Alhoniemi E, Parhankangas J. Self-organizing map in Matlab: the SOM Toolbox. Paper presented at: Proceedings of the Matlab DSP conference; November, 1999; Espoo, Finland.
140. Ramsay JO. *Functional data analysis*. 2 ed. New York: Springer-Verlag; 2006.
141. Yin S, Ding SX, Haghani A, Hao H, Zhang P. A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *J. Process Control.* 2012;22(9):1567-1581.